

## 《数据库原理》课程 实验报告

# 上机实验：Employees 数据库\_2

### 一、实验目的

1. 通过上机练习掌握关系数据库编程技术。

### 二、实验原理

1. 用 SQL 进行高级数据库操作。
2. 用 JDBC 编程连接数据库并进行操作。

### 三、实验内容

5.19 触发器实验：实现自动审计日志。

修改部门名称时，将数据库用户登录名、修改时间、部门编号、部门名称的旧值、部门名称的新值记录到 departments\_copy\_log 表中。

(1) 准备

执行语句

```
CREATE TABLE departments_copy AS (SELECT * FROM departments);
```

将部门表的所有行复制到新表 departments\_copy 中。

(2) 建立 departments\_copy\_log 表

执行语句

```
CREATE TABLE departments_copy_log
```

```
(
    log_id          SERIAL,    -- 日志流水编号 (自增)
    login_name      VARCHAR(256), -- 登录名
    update_date     TIMESTAMP WITH TIME ZONE, -- 修改时间
    dept_no         CHAR(4),    -- 部门编号
    dept_name_old   VARCHAR(40), -- 部门名称的旧值
    dept_name_new   VARCHAR(40), -- 部门名称的新值
    CONSTRAINT departments_copy_log_pk PRIMARY KEY(log_id)
);
```

(3) 使用网络资源学习 MariaDB/MySQL 的触发器语法。

(4) 编写触发器，实现修改部门名称时，将数据库用户登录名、修改时间、部门编号、部门名称的旧值、部门名称的新值记录到 departments\_copy\_log 表中。

```
CREATE or replace function addLog() returns trigger
```

```
language 'plpgsql'
```

```
AS $$
```

```
BEGIN
```

```
    insert into departments_copy_log(login_name, update_date, dept_no, dept_name_old, dept_name_new)
    values (session_user, now(), new.dept_no, old.dept_name, new.dept_name);
```

学号: 3019244266 姓名: 李润泽 日期: 2021 年 6 月 25 日 地点: 北洋园校区

```
return new;  
END;  
$$
```

```
create trigger log_department_change after update on departments_copy  
for each row  
execute procedure addLog();
```

(5) 执行 UPDATE 语句

```
UPDATE departments_copy  
SET dept_name = CONCAT(dept_name, ' Dept')  
WHERE dept_no = 'd005';
```

(5) 执行查询语句

```
SELECT * FROM departments_copy;
```

查看 UPDATE 对于 departments\_copy 表的修改。

(6) 执行查询语句

```
SELECT * FROM departments_copy_log;
```

查看触发器的作用, 是否实现了题目要求的审计日志的自动记录?

5.20 建立财务部门 (名称为 Finance) 的员工视图 finance\_employees\_view, 要求包括员工编号、员工姓名、性别、出生日期和入职日期。

```
create view finance_employees_view as select  
emp_no,first_name,last_name,birth_date,hire_date from employees;
```

执行语句:

```
SELECT * FROM finance_employees_view LIMIT 10;
```

查询视图 finance\_employees\_view 的前 10 行, 返回结果为:

	emp_no integer	first_name character varying (14)	last_name character varying (16)	birth_date date	hire_date date
1	10042	Magy	Stamatou	1956-02-26	1993-03-21
2	10050	Yinghua	Dredge	1958-05-21	1990-12-25
3	10059	Alejandro	McAlpine	1953-09-19	1991-06-26
4	10080	Premal	Baek	1957-12-03	1985-11-19
5	10132	Ayakannu	Skrikant	1956-12-15	1994-10-30
6	10144	Marla	Brendel	1959-06-17	1985-10-14
7	10146	Chenyl	Syang	1959-01-12	1988-06-28
8	10147	Kazuhito	Encarnacion	1964-10-13	1986-08-21
9	10165	Miyeon	Macedo	1960-06-16	1988-05-17
10	10173	Shrikanth	Mahmud	1962-10-28	1992-03-21

5.21 索引的作用。



(1) 查询员工“Peternela Anick”的全部属性, 记录查询执行时间。(提示: 通过学习 MariaDB/MySQL 文档, 使用 SET profiling = 1; 语句在查询之前打开记录时间功能, 使用 SET profiling = 0; 语句在查询之后关闭记录时间功能, 使用 SHOW PROFILES; 语句查看查询执行时间)

(2) 使用 EXPLAIN 语句查看第(1)步中查询的查询执行计划。(提示: 通过 MariaDB/MySQL 文档学习 EXPLAIN 语法)

学号: 3019244266 姓名: 李润泽 日期: 2021 年 6 月 25 日 地点: 北洋园校区

EXPLAIN ANALYZE

select \* from employees where first\_name = 'Paternela' AND last\_name = 'Anick';

Data Output	Explain	Messages	Notifications
	<b>QUERY PLAN</b> text 		
1	Gather (cost=100...		
2	Workers Planned...		
3	Workers Launched...		
4	-> Parallel Seq S...		
5	Filter: (((first_...		
6	Rows Removed...		
7	Planning time: 0.1...		
8	Execution time: 56...		

在员工表 employees 的 first\_name 和 last\_name 属性列上建立索引。(提示: 通过 MariaDB/MySQL 文档学习 CREATE INDEX 语法)

```
create index employees_first_name_index on employees(first_name);
```

```
create index employees_last_name_index on employees(last_name);
```

(3) 再次执行第(1)步的查询, 记录查询执行时间 (提示: 使用 SHOW PROFILES; 语句查看查询执行时间)

使用 EXPLAIN 语句查看第(1)步中查询的查询执行计划, 与第(2)步给出的查询执行计划进行对比。

56.674ms

0.237ms

(4) 删除在 first\_name 和 last\_name 属性列上建立索引 (提示: 用 DROP INDEX 语句)。

```
drop index employees_first_name_index;
```

```
drop index employees_last_name_index;
```

5.22 索引与键。

(1) 执行查询:

```
SELECT d.dept_no, d.dept_name, e.emp_no, e.first_name, e.last_name, s.salary  
FROM departments AS d
```

```
    INNER JOIN dept_emp AS de ON d.dept_no=de.dept_no
```

```
    INNER JOIN employees AS e ON de.emp_no=e.emp_no
```

```
    INNER JOIN salaries AS s ON e.emp_no=s.emp_no
```

```
WHERE e.first_name='Paternela' AND e.last_name='Anick';
```

记录查询执行时间

学号： 3019244266 姓名： 李润泽 日期： 2021 年 6 月 25 日 地点： 北洋园校区

	dept_no character (4)	dept_name character varying (40)	emp_no integer	first_name character varying (14)	last_name character varying (16)	salary integer
1	d009	Customer Service	234348	Peternela	Anick	40000
2	d009	Customer Service	234348	Peternela	Anick	42701
3	d009	Customer Service	234348	Peternela	Anick	44499
4	d009	Customer Service	234348	Peternela	Anick	47187
5	d009	Customer Service	234348	Peternela	Anick	40521
6	d009	Customer Service	234348	Peternela	Anick	40521

✓ 已成功运行。查询运行时间总数: 131 毫秒。12 行受影响。

(2) 在员工表 employees 的 first\_name 和 last\_name 属性列上建立索引。再次执行第(1)步的查询，记录查询执行时间。

	dept_no character (4)	dept_name character varying (40)	emp_no integer	first_name character varying (14)	last_name character varying (16)	salary integer
1	d009	Customer Service	234348	Peternela	Anick	40000
2	d009	Customer Service	234348	Peternela	Anick	42701
3	d009	Customer Service	234348	Peternela	Anick	44499
4	d009	Customer Service	234348	Peternela	Anick	47187
5	d009	Customer Service	234348	Peternela	Anick	40521
6	d009	Customer Service	234348	Peternela	Anick	40521

✓ 已成功运行。查询运行时间总数: 64 毫秒。12 行受影响。

(3) 执行下列语句，删除外键：

-- drop foreign keys

ALTER TABLE salaries DROP FOREIGN KEY fk\_salaries\_employees;

ALTER TABLE titles DROP FOREIGN KEY fk\_titles\_employees;

ALTER TABLE dept\_emp DROP FOREIGN KEY fk\_dept\_emp\_employees;

ALTER TABLE dept\_emp DROP FOREIGN KEY fk\_dept\_emp\_departments;

ALTER TABLE dept\_manager DROP FOREIGN KEY fk\_dept\_manager\_employees;

ALTER TABLE dept\_manager DROP FOREIGN KEY fk\_dept\_manager\_departments;

再次执行第(1)步的查询，记录查询执行时间。

	dept_no character (4)	dept_name character varying (40)	emp_no integer	first_name character varying (14)	last_name character varying (16)	salary integer
1	d009	Customer Service	234348	Peternela	Anick	40000
2	d009	Customer Service	234348	Peternela	Anick	42701
3	d009	Customer Service	234348	Peternela	Anick	44499
4	d009	Customer Service	234348	Peternela	Anick	47187
5	d009	Customer Service	234348	Peternela	Anick	40521
6	d009	Customer Service	234348	Peternela	Anick	40521

✓ 已成功运行。查询运行时间总数: 91 毫秒。12 行受影响。

(4) 执行下列语句，删除主键：

-- drop primary keys

ALTER TABLE employees DROP PRIMARY KEY;

ALTER TABLE departments DROP PRIMARY KEY;

ALTER TABLE dept\_emp DROP PRIMARY KEY;

ALTER TABLE salaries DROP PRIMARY KEY;

再次执行第(1)步的查询，记录查询执行时间。

学号： 3019244266 姓名： 李润泽 日期： 2021 年 6 月 25 日 地点： 北洋园校区

	dept_no character (4)	dept_name character varying (40)	emp_no integer	first_name character varying (14)	last_name character varying (16)	salary integer
1	d009	Customer Service	234348	Petemela	Anick	40000
2	d009	Customer Service	234348	Petemela	Anick	42701
3	d009	Customer Service	234348	Petemela	Anick	44499
4	d009	Customer Service	234348	Petemela	Anick	47187
5	d009	Customer Service	234348	Petemela	Anick	48501
6	d009	Customer Service	234348	Petemela	Anick	48501

已成功运行。查询运行时间总数: 3 秒 127 毫秒。12 行受影响。

(5) 执行下列语句，删除员工表 employees 的 first\_name 和 last\_name 属性列上建立的索引：

-- drop index

drop index employees first\_name index;

drop index employees last\_name index;

再次执行第(1)步的查询，记录查询执行时间。

这一段删掉

(5) 执行下列语句恢复主键：

-- add primary keys







ALTER TABLE employees ADD CONSTRAINT pk\_employees PRIMARY KEY(emp\_no);

ALTER TABLE departments ADD CONSTRAINT pk\_departments PRIMARY KEY(dept\_no);

ALTER TABLE dept\_emp ADD CONSTRAINT pk\_dept\_emp PRIMARY KEY(emp\_no, dept\_no);

ALTER TABLE salaries ADD CONSTRAINT pk\_salaries PRIMARY KEY(emp\_no, from\_date);

再次执行第(1)步的查询，记录查询执行时间。

数据输出	解释	消息	通知			
 dept_no character (4)	 dept_name character varying (40)	 emp_no integer	 first_name character varying (14)	 last_name character varying (16)	 salary integer	
1	d009	Customer Service	234348	Petemela	Anick	40000
2	d009	Customer Service	234348	Petemela	Anick	42701
3	d009	Customer Service	234348	Petemela	Anick	44499
4	d009	Customer Service	234348	Petemela	Anick	47187
5	d009	Customer Service	234348	Petemela	Anick	48501
6	d009	Customer Service	234348	Petemela	Anick	48501

✔ 已成功运行。查询运行时间总数: 84 毫秒。12 行受影响。

(6) 执行下列语句恢复外键：

-- add foreign keys

ALTER TABLE salaries ADD CONSTRAINT fk\_salaries\_employees FOREIGN KEY (emp\_no)  
REFERENCES employees(emp\_no);

ALTER TABLE titles ADD CONSTRAINT fk\_titles\_employees FOREIGN KEY (emp\_no)  
REFERENCES employees(emp\_no);

ALTER TABLE dept\_emp ADD CONSTRAINT fk\_dept\_emp\_employees FOREIGN KEY (emp\_no)  
REFERENCES employees(emp\_no);

ALTER TABLE dept\_emp ADD CONSTRAINT fk\_dept\_emp\_departments FOREIGN KEY  
(dept\_no) REFERENCES departments(dept\_no);

ALTER TABLE dept\_manager ADD CONSTRAINT fk\_dept\_manager\_employees FOREIGN KEY  
(emp\_no) REFERENCES employees(emp\_no);

ALTER TABLE dept\_manager ADD CONSTRAINT fk\_dept\_manager\_departments FOREIGN  
KEY (dept\_no) REFERENCES departments(dept\_no);

再次执行第(1)步的查询，记录查询执行时间。

学号： 3019244266 姓名： 李润泽 日期： 2021 年 6 月 25 日 地点： 北洋园校区

dept_no	dept_name	emp_no	first_name	last_name	salary
character (4)	character varying (40)	integer	character varying (14)	character varying (16)	integer
1 d009	Customer Service	234348	Petemela	Anick	40000
2 d009	Customer Service	234348	Petemela	Anick	42701
3 d009	Customer Service	234348	Petemela	Anick	44499
4 d009	Customer Service	234348	Petemela	Anick	47187
5 d009	Customer Service	234348	Petemela	Anick	40000
6 d009	Customer Service	234348	Petemela	Anick	40000

已成功运行。查询运行时间总数: 93 毫秒。12 行受影响。

(7) 在员工表 employees 的 first\_name 和 last\_name 属性列上建立索引。再次执行第(1)步的查询，记录查询执行时间。

dept_no	dept_name	emp_no	first_name	last_name	salary
character (4)	character varying (40)	integer	character varying (14)	character varying (16)	integer
1 d009	Customer Service	234348	Petemela	Anick	40000
2 d009	Customer Service	234348	Petemela	Anick	42701
3 d009	Customer Service	234348	Petemela	Anick	44499
4 d009	Customer Service	234348	Petemela	Anick	47187
5 d009	Customer Service	234348	Petemela	Anick	40000
6 d009	Customer Service	234348	Petemela	Anick	40000

已成功运行。查询运行时间总数: 86 毫秒。12 行受影响。

### 5.23 存储过程。

(1) 通过 MariaDB/MySQL 文档，学习 CREATE PROCEDURE 语句的语法。

(2) 创建存储过程 calc\_avg\_salary\_for\_emp\_no,

其有两个参数：emp\_no\_in 输入型，指定员工编号；

avg 输出型，保存工资平均值。

该存储过程的功能是：计算编号为 emp\_no\_in 的员工在工资表 salaries 中的工资数额的平均值，并将该平均值保存到@avg 中。

```
create or replace function calc_avg_salary_for_emp_no(in emp_no_in INTEGER, out avg FLO
AT) returns FLOAT
```

```
language 'plpgsql'
```

```
AS $$
```

```
BEGIN
```

```
    select avg(salary)
```

```
    into avg
```

```
    from salaries
```

```
    where emp_no = emp_no_in;
```

```
END;
```

```
$$
```

(3) 执行存储过程 calc\_avg\_salary\_for\_emp\_no，执行下列语句，调用存储过程并输出结果：

```
CALL calc_avg_salary_for_emp_no(10002, @avg_salary);
```

```
SELECT @avg_salary;
```

执行结果为：

Data Output Explain Messages Notifications

注意： 68854.5

D0

(4) 使用 DROP PROCEDURE 语句，删除存储过程 calc\_avg\_salary\_for\_emp\_no

Data Output	Explain	Messages	Notifications
DROP FUNCTION			
Query returned successfully in 42 msec.			

5.24 存储过程（函数、分支）。 (选做)

(1) 通过 MariaDB/MySQL 文档，学习 CREATE FUNCTION 语句的语法。创建存储过程 is\_manager,

其有一个参数：emp\_no\_in，指定员工编号；

返回值，BOOL 类型，返回 1 表示 emp\_no\_in 编号的员工是经理 (manager)，返回 0 表示 emp\_no\_in 编号的员工不是经理。

该函数的功能是：查询编号为 emp\_no\_in 的员工是否为经理 (在 dept\_manager 表中查询)。

(提示：使用 IF...ELSE 语句实现分支判断。)

(2) 执行存储过程 is\_manager，执行下列语句，调用函数并输出返回结果：

SELECT is\_manager(110022) AS is\_manager;

执行结果为：

is_manager
1

(3) 执行存储过程 is\_manager，执行下列语句，调用函数并输出返回结果：

SELECT is\_manager(100002) AS is\_manager;

执行结果为：

is_manager
0

(4) 使用 DROP FUNCTION 语句，删除函数 is\_manager

5.25 存储过程（游标、循环）。 (选做)

(1) 创建存储过程 calc\_avg\_and\_var\_salary\_for\_emp\_no,

其有三个参数：emp\_no\_in，输入型，指定员工编号；

avg，REAL 类型，输出型，返回 emp\_no\_in 指定的员工的工资数额的平均值。

var，REAL 类型，输出型，返回 emp\_no\_in 指定的员工的资数额的方差。

该函数的功能是：计算编号为 emp\_no\_in 的员工的工资数额的平均值和方差。

(提示：使用游标获取指定 emp\_no\_in 的员工的每条工资记录中的工资数额。使用 WHILE 语句编写循环。)

(3) 执行存储过程 calc\_avg\_and\_var\_salary\_for\_emp\_no，执行下列语句，调用存储过程并输出结果：

CALL calc\_avg\_and\_var\_salary\_for\_emp\_no(10002, @avg\_salary, @var\_salary);

SELECT @avg\_salary AS avg\_salary, @var\_salary AS var\_salary;

执行结果为：

avg_salary	var_salary
68854.5	7165175.583333015

学号: 3019244266 姓名: 李润泽 日期: 2021 年 6 月 25 日 地点: 北洋园校区

(4) 使用 DROP PROCEDURE 语句,

删除存储过程 calc\_avg\_and\_var\_salary\_for\_emp\_no

#### 5.26 JDBC 编程。

(选做)

编写 Java 程序, 通过 JDBC 连接 MariaDB/MySQL 中的 Employees 数据库, 实现对 departments 表中记录的查询 (SELECT)、添加 (INSERT)、修改 (UPDATE) 和删除 (DELETE)。

(1) JDBC驱动程序: MariaDB Client Library for Java Applications中的 mariadb-java-client-1.1.7.jar (课程主页上下载)。

(2) Eclipse项目JDBCEmployees (课程主页上下载)。

(3) 编写代码, 实现DepartmentsDAO类中的TODO部分。



#### 四、实验总结

(300 字以上)

Lab04 结束之后，我感触很多，在这几天的实验中，我对数据库的理论和实操知识掌握了很多。期间我学到了很多，包括索引和键对查询效率的影响，通过建立索引和主键大大提高了查询效率，即主键和索引的加入都可以提高查询效率，而外键则正好相反。

从实验中让我更明白一些知识，表是数据最重要的一个数据对象，表的创建好坏直接关系到数据库的成败，表的内容是越具体越好，但是也不能太繁琐，以后在实际应用中多使用表，对表的规划和理解就会越深刻。

在本次上机实验中，我巩固了关系数据模型及其基本概念；巩固了关系数据库语言 SQL；巩固了关系数据库设计方法。全程采用 PostgreSQL 数据库作为实验用 DBMS，此后，成功用 SQL 创建数据库模式（其中包括数据表的建立，主键，外键的声明）。随后，用 copyfrom 语句将数据批量装载到数据库中，并对导入数据进行检查。确认数据正确无误后，用 SQL 进行查询和更新操作（主要涉及查询，删除，插入等操作），其中灵活运用了 GROUP BY, COUNT(\*), SELECT FROM WHERE, DELETE, INSERT 等语句。有许多语句的使用方式是实验指导中没有给出的，需要我自行学习，而我也在经历了种种失败之后终于成功，提高了我自学的能力和对 SQL 语句的了解和认识。