学号：3012216083　　姓名：王雨朦　　日期：2014 年 11 月 24 日　　地点：综合实验楼

《数据库原理》课程　实验报告

上机实验：Employees 数据库（1）

一、实验目的

1. 通过上机练习巩固关系数据库设计方法。
2. 通过上机练习巩固关系数据库语言 SQL。

二、实验原理

3. 采用 MariaDB 或 MySQL 数据库作为实验用 DBMS。
4. 用 E/R 图建立数据库的概念模型。
5. 将 E/R 模型转换为关系模型。
6. 用 SQL 创建数据库模式。
7. 将数据批量装载到数据库中。
8. 用 SQL 进行查询和更新操作。

三、实验内容

1. 用户需求。

(1) 某公司为管理员工相关数据需要设计名为 Employees 的数据库。该数据库中要管理的数据包括：员工数据（employees）、职称数据（titles）、工资数据（salaries）、部门数据（departments）等。

(2) 员工数据包括：员工编号（emp_no）、出生日期（birth_date）、名字（first_name）、姓氏（last_name）、性别（gender）、入职日期（hire_date）。

(3) 职称数据包括：职称名称（title）、起始时间（from_date）、终止时间（to_date）。一条职称数据记录了某员工从起始时间到终止时间这个时间段内的职称名称。

(4) 工资数据包括：工资数额（salary）、起始时间（from_date）、终止时间（to_date）。一条工资数据记录了某员工从起始时间到终止时间这个时间段内的工资数额。

(5) 部门数据包括：部门编号（dept_no）、部门名称（dept_name）。

(6) 部门和员工间的关系 1（dept_emp）：一个部门下属有多名员工，一名员工可隶属于多个部门。需要记录某员工为某部门工作的起始时间和终止时间。

(7) 部门和员工间的关系 2（dept_manager）一个部门有多位经理（不用区分

正副职），经理也是一名员工，一名员工可同时担任多个部门的经理。需要记录某员工担任某部门经理的起始时间和终止时间。

2. 分析用户需求，画出 Employees 数据库的 E/R 模型图。

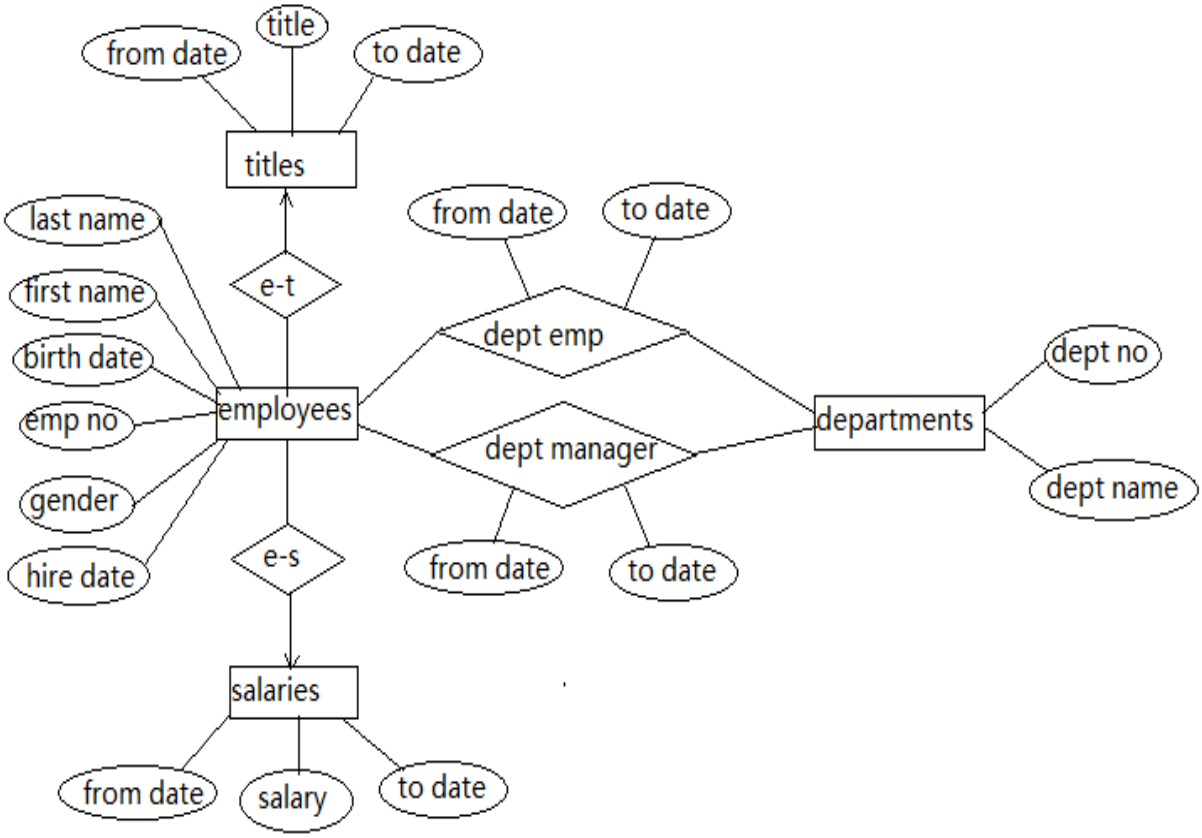

图 2.1: Employees 的 E/R 模型图

3. 将 E/R 模型转换为关系模型，用 SQL 创建关系表，写出 CREATE TABLE 语句。

要求：用 CONSTRAINT 关键字建立有名称的主键和外键约束。

主键名称格式为：pk_表名

外键名称格式为：fk_本表名_引用表名

```
create table employees(
    emp_no char(20),
    birth_date char(20),
    first_name char(20),
    last_name char(20),
    gender char(10),
    hire_date char(20),
    constraint pk_employees primary key (emp_no)
);
```

```
create table departments(
    dept_no char(20),
    dept_name char(20),
    constraint pk_departments primary key(dept_no)
);

create table dept_emp(
    emp_no char(20),
    dept_no char(20),
    from_date char(20),
    to_date char(20),
    constraint fk_dept_emp_departments foreign key (dept_no)
    references departments(dept_no),
    constraint fk_dept_emp_employees foreign key (emp_no) references
    employees(emp_no)
);

create table dept_manager(
    dept_no char(20),
    emp_no char(20),
    from_date cchar(20),
    to_date char(20),
    constraint fk_dept_manager_departments foreign key (dept_no)
    references departments(dept_no),
    constraint fk_dept_manager_employees foreign key (emp_no)
    references employees(emp_no)
);
create table titles(
    emp_no char(20),
    titles char(20),
    from_date char(20),
    to_date char(20),
    constraint fk_titles_employees foreign key (emp_no) references
    employees (emp_no)
);

create table salaries(
    emp_no char(20),
    salary char(20),
    from_date char(20),
    to_date char(20),
    constraint fk_salaries_employees foreign key (emp_no) references
    employees (emp_no)
);
```

4. 将提供的示例数据导入到已创建的表中。

数据文件说明：

data_employees.txt	员工数据
data_departments.txt	部门数据
data_dept_emp.txt	部门员工关系数据
data_dept_manager.txt	部门经理关系数据
data_salaries.txt	工资数据
data_titles.txt	职称数据

使用 MariaDB/MySQL 提供的批量导入数据的语句 LOAD DATA INFILE。

（关于 LOAD DATA INFILE 语法解释，请自己查询 MariaDB/MySQL 文档）

导入之后的结果：

employees 表	300024	行数据
departments 表	9	行数据
dept_emp 表	331603	行数据
dept_manager 表	24	行数据
titles 表	443308	行数据
salaries 表	2844047	行数据

```
-- 这里路径名必须是正斜杠
load data infile "C:/data_employees.txt" into table employees
fields terminated by ','
lines terminated by '\n';

load data infile "C:/data_departments.txt" into table departments
fields terminated by ','
lines terminated by '\n';

load data infile "C:/data_dept_emp.txt" into table dept_emp
fields terminated by ','
lines terminated by '\n';
load data infile "C:/data_dept_manager.txt" into table dept_manager
fields terminated by ','
lines terminated by '\n';
load data infile "C:/data_titles.txt" into table titles
fields terminated by ','
lines terminated by '\n';
load data infile "C:/data_salaries.txt" into table salaries
fields terminated by ','
lines terminated by '\n';
```

5. 按照下列查询要求编写 SQL 语句。

5.1 返回前 10 行员工数据。

（提示：用 LIMIT 关键字，具体用法查文档）

```
SELECT * FROM employees LIMIT 0,10;
```

查询执行结果：

emp_no	birth_date	first_name	last_na...	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
10009	1952-04-19	Sumant	Peac	F	1985-02-18
10010	1963-06-01	Duangka...	Piveteau	F	1989-08-24

5.2 查询 first_name 为 Peternela 且 last_name 为 Anick 的员工的编号、出生日期、性别和入职日期。

```
select emp_no,birth_date,gender,hire_date
from employees
where first_name='Peternela' and last_name='Anick';
```

查询执行结果：

emp_no	birth_date	gender	hire_date
234348	1961-07-16	M	1991-01-15

5.3 查询出生日期在 1961-7-15（包括）到 1961-7-20（包括）之间的员工的编号、姓名和出生日期。

```
select emp_no, first_name, last_name, birth_date
from employees
where birth_date>='1961-07-15' and birth_date<='1961-07-20';
```

查询执行结果：（386 行数据返回）

5.4 查询所有 first_name 中含有前缀 Peter 或 last_name 中含有前缀 Peter 的员工数据（返回所有列）。

```
select *
from employees
where first_name like 'Peter%' or last_name like 'Peter%';
```

查询执行结果：（771 行数据返回）

5.5 查询工资数额的最大值，并将查询结果的列名命名为 max_salary。

```
select max(salary) as max_salary
from salaries;
```

查询执行结果：

max_salary
158220

5.6 查询部门编号及相应部门的员工人数，并按照部门编号由小到大的顺序排序（将员工人数列命名为 dept_emp_count）。

```
select dept_no, count(emp_no) as dept_emp_count
from dept_emp
group by dept_no;
```

查询执行结果：

dept_no	dept_emp_count
d001	20211
d002	17346
d003	17786
d004	73485
d005	85707
d006	20117
d007	52245
d008	21126
d009	23580

5.7 查询员工“Peternela Anick”的员工编号、所在部门编号和在该部门的工作起始时间。

```
select dept_emp.emp_no, dept_no, from_date
from employees join dept_emp on employees.emp_no=dept_emp.emp_no
where first_name='Peternela' and last_name='Anick';
```

查询执行结果：

emp_no	dept_no	from_date
234348	d009	1991-01-15

5.8 查询姓名相同的员工 x 和员工 y 的编号和姓名（只列出前 10 行结果）。

```
select A.emp_no, B.emp_no, A.first_name, B.last_name
from employees A, employees B
where A.last_name=B.last_name and  A.first_name=B.first_name and
A.emp_no<B.emp_no
limit 0,10;
```

查询执行结果：

emp_no	emp_no	first_name	last_name
10001	55649	Georgi	Facello
10029	48235	Otmar	Herbst
10029	73878	Otmar	Herbst
10036	435453	Adamantios	Portugali
10045	495302	Moss	Shanbhogue
10053	109174	Sanjiv	Zschoche
10071	246572	Hisao	Lipner
10074	45973	Mokhtar	Bernatsky
10074	432734	Mokhtar	Bernatsky
10075	25260	Gao	Dolinsky

5.9 查询姓名为“Margo Anily”的员工编号和出生日期为“1959-10-30”入职日期为“1989-09-12”的员工编号的并集。

```
select emp_no
from employees
where (first_name='Margo' and last_name='Anily') or
(birth_date='1959-10-30' and hire_date='1989-09-12');
```

查询执行结果：

emp_no
16601
89561

5.10 查询员工“Margo Anily”所在的部门的名称（要求用子查询实现）。

```
select dept_name
from departments
where dept_no in
  (select dept_no
  from dept_emp
  where emp_no in
    (select emp_no
    from employees
    where first_name='Margo' and last_name='Anily')
  );
```

查询执行结果：

dept_name
Production

5.11 要求用 JOIN…ON 连接语法实现查询 5.10。

```
select dept_name
from departments natural join dept_emp  natural join employees
where first_name='Margo' and last_name='Anily';
```

查询执行结果：

dept_name
Production

5.12 查询在全部部门中工作过的员工的编号和姓名（提示：用 NOT EXISTS 连接的子查询）。

```
select employees.emp_no, first_name, last_name
from employees natural join dept_emp
group by employees.emp_no
having count(*)=(
    select count(*)
    from departments
);
```

查询执行结果：

emp_no	first_name	last_name

5.13 查询员工人数大于等于 50000 的部门编号、部门名称和部门员工人数，按照部门编号由小到大的顺序排序（将部门员工人数列命名为 dept_emp_count）。

```
select dept_emp.dept_no, dept_name, count(*) as dept_emp_count
from dept_emp natural join departments
group by dept_emp.dept_no
having count(*)>=50000;
```

查询执行结果：

dept_no	dept_name	dept_emp_count
d004	Production	73485
d005	Development	85707
d007	Sales	52245

5.14 在员工表中添加一行记录：

(10000, 1981-10-1, Jimmy, Lin, M, 2011-12-8)

```
insert into employees
value (10000, 1981-10-01, Jimmy, Lin, M, 2011-12-08);
```

查询执行结果：

(1 行受影响)

5.15 将 5.14 添加的员工记录的 first_name 属性值修改为 Jim。

```
update employees
set first_name='Jim'
where emp_no=10000;
```

查询执行结果：

(1 行受影响)

5.16 删除 5.14 添加的员工记录。

```
delete
from employees
where emp_no=10000;
```

查询执行结果：

(1 行受影响)

5.17 在员工表中添加一行记录：

(10001, 1981-10-1, Jimmy, Lin, M, 2011-12-8)，观察执行输出结果。

```
insert into employees
value (10001, '1981-10-01', 'Jimmy', 'Lin', 'M', '2011-12-08');
```

查询执行结果：

SQL 错误（1062）：Duplicate entry '10001' for key 1

这是由于插入的数据重复导致的错误。

5.18 删除编号为 10001 的员工，观察执行输出结果。

```
delete
from employees
where emp_no=10001;
```

查询执行结果：

SQL 错误（1451）：Cannot delete or update a parent row: a foreign key constraint fails (`employees/dept_emp`, CONSTRAINT `fk_dept_emp_employees` FOREIGN KEY (`emp_no`) REFERENCES `employees` (`emp_no`))

这是由于要删除的属性是另外一个表的外键，删除时会自动检查外键。

四、 实验总结

这次实验内容是对以前学习的数据库知识的一个升华。以前我们只是简单的学习了一些建立数据、建立表、向表中简单的插入数据和在表中的一些查询。而这次的实验内容是教会我们从学习如何使用数据库语句到设计一个简单的数据库。

关于如何设计数据库，我有了自己的思考并且通过查询资料和总结得到，数据库的简单设计过程为：需求分析->概念结构设计（画出 E/R 物理模型图）->逻辑结构设计（将 E/R 图转换为关系模型）->创建数据库。

此外，还学习到几点：

1. 创建数据库之后，数据并不是简单的自己一条条插入，而是通过.txt 文件导入相对多的数据。更接近现实生活中实际的数据库建立。
2. 通过对不同查询语句的使用，体会到 SQL 语句是一个灵活多变的实用体系，它针对不同的需求设计了不同的语句。
3. 通过对最后两道报错语句的尝试，我们也学习到重复插入是不被数据库接受的，并且数据库是一个严谨的体系，如果有外键关联的属性也不能够随便删除。

通过这次上机实验，巩固了我以前学习到的数据库相关知识，并且将理论转化为了实践，加深了我对数据库知识和 SQL 语句的认识。