

## 实习二：数据库约束设计

成员：陈嘉2000013094 马佳媛2000013121 黄粤2000013095

In [1]:

```
%load_ext sql
import pymysql
pymysql.install_as_MySQLdb()
%sql mysql://stu2000013121:stu2000013121@162.105.146.37:53306
%sql use stu2000013121;
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
```

```
0 rows affected.
```

Out[1]:

```
[]
```

In [2]:

```
#查看所有表
%sql show tables;
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
```

```
9 rows affected.
```

Out[2]:

**Tables\_in\_stu2000013121**

bank_card
client
cost
department
employees
finances_product
fund
insurance
property

本次实习的目标是请同学们体验如何在数据库中利用各种手段完成数据库约束设计，以及如何使用触发器完成类似物化视图的数据一致性维护。主要包括如下两个练习：

## 练习一 约束设计

### 1、按照题目要求进行建表：

**employees(eno, ename, dno, salary, level, email)**

各属性约束如下：

**eno**：整型，主码；

**ename**：长度为100的字符型，非空；

**dno**：整型，非空，外码，与department表中的dno相互参照；

**salary**：整型，非空；

**level**：整型，非空；

**email**：长度为20的字符型，取值唯一，非空

In [3]:

```
%%sql
/*若多个表之间存在外键约束，不考虑顺序直接drop会失败，因此我们先取消外键约束的检查*/
set @@foreign_key_checks=0;

# 如果数据表已经存在则删除表
# 创建employees表
drop table if exists employees;

CREATE TABLE employees
(
    emp_eno INT PRIMARY KEY,
    emp_ename VARCHAR(100) NOT NULL,
    emp_dno INT NOT NULL,
    emp_salary INT NOT NULL,
    emp_level INT NOT NULL,
    emp_email VARCHAR(20) UNIQUE NOT NULL
);
set @@foreign_key_checks=1;
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[3]:

```
[]
```

**department(dno, dname, manager, budget)**

各属性约束如下：

**dno**：整型，主码；

**dname**：{'销售部'、'财务部'、'人事部'}之一，非空；

**manager**: 整型, 非空, 外码, 与employee表中的eno相互参照;

**budget**: 整型, 非空;

----- 表结构图 -----

In [4]:

```
%%sql
#若多个表之间存在外键约束, 不考虑顺序直接drop会失败, 因此我们先取消外键约束的检查
set @@foreign_key_checks=0;

# 如果数据表已经存在则删除表
drop table if exists department;

CREATE TABLE department
(
    dep_dno INT PRIMARY KEY,
    dep_dname ENUM('销售部', '财务部', '人事部') NOT NULL,
    dep_manager INT NOT NULL,
    dep_budget INT NOT NULL,
    dep_cost INT DEFAULT(0)
);
set @@foreign_key_checks=1;
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[4]:

[]

## 2、生成样例数据, 插入一些条目

In [5]:

```
employees;
department;

department(dep_dno, dep_dname, dep_manager, dep_budget, dep_cost) VALUES(1, '销售部', 7, 10000, 0);
department(dep_dno, dep_dname, dep_manager, dep_budget, dep_cost) VALUES(2, '财务部', 3, 20000, 0);
department(dep_dno, dep_dname, dep_manager, dep_budget, dep_cost) VALUES(3, '人事部', 5, 15000, 0);
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
0 rows affected.
0 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
```

Out[5]:

[]

In [6]:

%%sql

```
select * from department;
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
3 rows affected.
```

Out[6]:

dep_dno	dep_dname	dep_manager	dep_budget	dep_cost
1	销售部	7	10000	0
2	财务部	3	20000	0
3	人事部	5	15000	0

建立触发器，在每次插入新员工时更新对应的cost

In [7]:

%%sql

```
drop trigger if exists cost_by_now;
create trigger cost_by_now after insert
  on employees for each row
  update department set department.dep_cost=(select sum(emp_salary) from employees where emp_c
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
0 rows affected.
0 rows affected.
```

Out[7]:

[]

In [8]:

```
%%sql

INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(1, '员工1号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(2, '员工2号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(3, '员工3号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(4, '员工4号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(5, '员工5号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(6, '员工6号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(7, '员工7号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(8, '员工8号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(9, '员工9号
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES(10, '员工10号
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
```

Out[8]:

[]

In [9]:

```
%%sql

select * from employees;
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
10 rows affected.
```

Out[9]:

emp_eno	emp_ename	emp_dno	emp_salary	emp_level	emp_email
1	员工1号	1	1500	1	12345@qq.com
2	员工2号	1	1700	1	12346@qq.com
3	员工3号	3	3000	2	12347@qq.com
4	员工4号	1	3500	3	12348@qq.com
5	员工5号	3	4500	4	12349@qq.com
6	员工6号	2	5500	5	12325@qq.com
7	员工7号	2	1000	1	12335@qq.com
8	员工8号	3	1500	1	12315@qq.com

In [10]:

```
%%sql
```

```
select * from department;
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
3 rows affected.
```

Out[10]:

dep_dno	dep_dname	dep_manager	dep_budget	dep_cost
1	销售部	7	10000	6700
2	财务部	3	20000	12500
3	人事部	5	15000	9000

### 3、添加check约束限制取值范围

各约束描述如下：

**外键约束：**

**emp\_dno:**需要存在于department表中的dep\_dno中

**dep\_mananger:**需要存在于employees表中的emp\_eno中

**格式约束：**

**emp\_email:**正则表达式格式

**范围限制：**

**dep\_dname:**要求是{'销售部'、'财务部', '人事部'}之一

**emp\_level:**取值为1到5的整数

**emp\_salary:** emp\_salary的取值范围为[emp\_level \* 1000, emp\_level \* 1000 + 1000],同时一个部门所有员工的salary总和不能超过该部门的budget。

In [11]:

%%sql

```
ALTER TABLE employees ADD CONSTRAINT FK_emp_d FOREIGN KEY (emp_dno) REFERENCES department(dep_dno)
#添加外键约束, manager存在于eno中
ALTER TABLE department ADD CONSTRAINT FK_dep_m FOREIGN KEY (dep_manager) REFERENCES employees(emp_dno)
#email限制正则表达式格式
ALTER TABLE employees ADD CONSTRAINT CK_email CHECK (emp_email REGEXP '^[A-Za-z0-9\u4e00-\u9fa5]-[0-9]*$');
#level限制1-5之间
ALTER TABLE employees ADD CONSTRAINT CK_level CHECK (emp_level >= 1 and emp_level <= 5);
#salary为正整数, 且受到level的限制
ALTER TABLE employees ADD CONSTRAINT CK_salary CHECK (emp_salary >= 1000*emp_level and emp_salary <= 10000*emp_level);
#budget为正整数, 且高于已使用的预算值
ALTER TABLE department ADD CONSTRAINT CK_budget CHECK (dep_budget >= dep_cost);
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
10 rows affected.
3 rows affected.
10 rows affected.
10 rows affected.
10 rows affected.
3 rows affected.
```

Out[11]:

[]

## 4、对添加的约束条件进行验证

### (1)验证emp\_dno是否受dep\_dno约束:

输入超出范围的dno值会报错

In [12]:

```
%%sql

INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (11, '员工11号', 10, 1500, 1, 'EMP11@COMPANY.COM')

/usr/local/share/anaconda3/envs/py37/lib/python3.7/site-packages/pymysql/prot
ocol.py in raise_for_error(self)
    219         if DEBUG:
    220             print("errno =", errno)
--> 221         err.raise_mysql_exception(self._data)
    222
    223     def dump(self):

/usr/local/share/anaconda3/envs/py37/lib/python3.7/site-packages/pymysql/err.
py in raise_mysql_exception(data)
    141     if errorclass is None:
    142         errorclass = InternalError if errno < 1000 else OperationalE
error
--> 143     raise errorclass(errno, errval)

IntegrityError: (pymysql.err.IntegrityError) (1452, 'Cannot add or update a c
hild row: a foreign key constraint fails (`stu2000013121`.`employees`, CONSTR
AINT `FK_emp_d` FOREIGN KEY (`emp_dno`) REFERENCES `department` (`dep_dno`
`))')
[SQL: INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, em
```

dno范围内的值可以被正常插入

In [13]:

```
%sql

ERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (11, '员工11号', 10, 1500, 1, 'EMP11@COMPANY.COM')

* mysql://stu2000013121:***@162.105.146.37:53306
1 rows affected.
```

Out[13]:

[]

## (2)验证dep\_manager是否受emp\_eno约束:

输入超出范围的dep\_manager值会报错



In [14]:

```
%%sql

INSERT INTO department(dep_dno, dep_dname, dep_manager, dep_budget, dep_cost) VALUES(4, '销售部', 15, 100000)

/usr/local/share/anaconda3/envs/py37/lib/python3.7/site-packages/pymysql/protocol.py in raise_for_error(self)
    219         if DEBUG:
    220             print("errno =", errno)
--> 221         err.raise_mysql_exception(self._data)
    222
    223     def dump(self):

/usr/local/share/anaconda3/envs/py37/lib/python3.7/site-packages/pymysql/err.py in raise_mysql_exception(data)
    141     if errorclass is None:
    142         errorclass = InternalError if errno < 1000 else OperationalError
--> 143     raise errorclass(errno, errval)

IntegrityError: (pymysql.err.IntegrityError) (1452, 'Cannot add or update a child row: a foreign key constraint fails (`stu2000013121`.`department`, CONSTRAINT `FK_dep_m` FOREIGN KEY (`dep_manager`) REFERENCES `employees` (`emp_eno`))')
[SQL: INSERT INTO department(dep_dno, dep_dname, dep_manager, dep_budget, dep_cost)
```

**(3)验证dep\_dname是否受取值范围约束:**

输入超出范围的dep\_dname值会报错

In [15]:

```
%%sql

INSERT INTO department(dep_dno, dep_dname, dep_manager, dep_budget, dep_cost) VALUES(4, '随便部', 2, 100000)

/usr/local/share/anaconda3/envs/py37/lib/python3.7/site-packages/pymysql/protocol.py in raise_for_error(self)
    219         if DEBUG:
    220             print("errno =", errno)
--> 221         err.raise_mysql_exception(self._data)
    222
    223     def dump(self):

/usr/local/share/anaconda3/envs/py37/lib/python3.7/site-packages/pymysql/err.py in raise_mysql_exception(data)
    141     if errorclass is None:
    142         errorclass = InternalError if errno < 1000 else OperationalError
--> 143     raise errorclass(errno, errval)

DataError: (pymysql.err.DataError) (1265, "Data truncated for column 'dep_dname' at row 1")
[SQL: INSERT INTO department(dep_dno, dep_dname, dep_manager, dep_budget, dep_cost)
```

#### (4)验证emp\_level是否受取值范围约束:

输入超出范围的emp\_level值会报错

In [16]:

```
%%sql
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (12, '员工12号', 1, 1700, 11, '22346@qq.com');
```

\* mysql://stu2000013121:\*\*\*@162.105.146.37:53306  
(pymysql.err.OperationalError) (3819, "Check constraint 'CK\_level' is violated.")  
[SQL: INSERT INTO employees(emp\_eno, emp\_ename, emp\_dno, emp\_salary, emp\_level, emp\_email) VALUES (12, '员工12号', 1, 1700, 11, '22346@qq.com');]  
(Background on this error at: <https://sqlalche.me/e/14/e3q8>) (<https://sqlalche.me/e/14/e3q8>)

#### (5)验证emp\_salary是否受取值范围约束:

输入超出范围的emp\_salary值会报错

In [17]:

```
%%sql
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (13, '员工13号', 1, 1700, 2, '32346@qq.com');
```

\* mysql://stu2000013121:\*\*\*@162.105.146.37:53306  
(pymysql.err.OperationalError) (3819, "Check constraint 'CK\_salary' is violated.")  
[SQL: INSERT INTO employees(emp\_eno, emp\_ename, emp\_dno, emp\_salary, emp\_level, emp\_email) VALUES (13, '员工13号', 1, 1700, 2, '32346@qq.com');]  
(Background on this error at: <https://sqlalche.me/e/14/e3q8>) (<https://sqlalche.me/e/14/e3q8>)

#### (6)验证emp\_email是否符合正则表达式:

输入格式不符的emp\_email值会报错

In [18]:

```
%%sql
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (14, '员工14号', 1, 1700, 1, '32346');
```

\* mysql://stu2000013121:\*\*\*@162.105.146.37:53306  
(pymysql.err.OperationalError) (3819, "Check constraint 'CK\_email' is violated.")  
[SQL: INSERT INTO employees(emp\_eno, emp\_ename, emp\_dno, emp\_salary, emp\_level, emp\_email) VALUES (14, '员工14号', 1, 1700, 1, '32346');]  
(Background on this error at: <https://sqlalche.me/e/14/e3q8>) (<https://sqlalche.me/e/14/e3q8>)

**(7)验证一个部门的员工salary总和是否不能超过该部门的budget:**

输入员工salary超出时会报错

In [19]:

```
%%sql
```

```
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (15, '员工15号', 3, 1700, 1, '92346@qq.com');
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (16, '员工16号', 3, 1700, 1, '92346@qq.com');
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (17, '员工17号', 3, 1700, 1, '92346@qq.com');
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (18, '员工18号', 3, 1700, 1, '92346@qq.com');
INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (19, '员工19号', 3, 1700, 1, '92346@qq.com');
```

```
* mysql://stu2000013121:***@162.105.146.37:53306
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
(pymysql.err.OperationalError) (3819, "Check constraint 'CK_budget' is violated.")
[SQL: INSERT INTO employees(emp_eno, emp_ename, emp_dno, emp_salary, emp_level, emp_email) VALUES (19, '员工19号', 3, 1700, 1, '92346@qq.com');]
(Background on this error at: https://sqlalche.me/e/14/e3q8) (https://sqlalche.me/e/14/e3q8)
```

In [20]:

```
%%sql
select * from employees;
```

\* mysql://stu2000013121:\*\*\*@162.105.146.37:53306  
15 rows affected.

Out[20]:

emp_eno	emp_ename	emp_dno	emp_salary	emp_level	emp_email
1	员工1号	1	1500	1	12345@qq.com
2	员工2号	1	1700	1	12346@qq.com
3	员工3号	3	3000	2	12347@qq.com
4	员工4号	1	3500	3	12348@qq.com
5	员工5号	3	4500	4	12349@qq.com
6	员工6号	2	5500	5	12325@qq.com
7	员工7号	2	1000	1	12335@qq.com
8	员工8号	3	1500	1	12315@qq.com
9	员工9号	2	2500	2	12375@qq.com
10	员工10号	2	3500	3	12395@qq.com
11	员工11号	1	1700	1	22346@qq.com
15	员工15号	3	1000	1	52346@qq.com
16	员工16号	3	1000	1	62346@qq.com
17	员工17号	3	2000	2	72346@qq.com
18	员工18号	3	2000	2	82346@qq.com

In [21]:

```
%%sql
select * from department;
```

\* mysql://stu2000013121:\*\*\*@162.105.146.37:53306  
3 rows affected.

Out[21]:

dep_dno	dep_dname	dep_manager	dep_budget	dep_cost
1	销售部	7	10000	8400
2	财务部	3	20000	12500
3	人事部	5	15000	15000

In [ ]:

