

# 数据库系统实验十

## 18340013 陈琮昊

### 一、实验目的：

学习用户自定义约束，并实践用户完整性，利用短语 NOT NULL, UNIQUE, CHECK 保证用户定义完整性。

### 二、实验内容：

创建 worker 表，进行实践用户完整性。

### 三、实验题目：

注：以下练习均在 worker 表上进行（请按照示例中 worker 建表）

- 1.加入约束 U3，令 sage 值大于等于0。
- 2.演示插入违反约束 U3 的记录。
- 3.演示插入不违反约束 U3 的记录。
- 4.加入约束 U4，令 sage 值<0，观察执行是否成功，分析原因。
- 5.加入规则 R2，确保插入的记录的 sage 值在1到100之间，并绑定到 sage 属性。
- 6.演示插入违反 R2 的记录。
- 7.解除规则 R2 的绑定，并重复6的操作。
- 8.已知示例三中已插入 sage 为38的记录，加入规则 R3，令 sage 大于50。观察加入规则 R3 的操作是否能成功。

### 四、实验过程与结果：

- 1.添加约束 U3 代码与结果如下：

```
alter table Worker  
add constraint U3 check(sage>=0)
```



命令已成功完成。

2.插入一个 `sage=-30` 的数据可以看到失败，因为违反了约束  $sage \geq 0$ :

```
insert into Worker values('00004','陈','1',-30,'研发部')
```



消息 547，级别 16，状态 0，第 1 行

INSERT 语句与 CHECK 约束“U3”冲突。该冲突发生于数据库“School”，表“dbo.Worker”，column ‘Sage’。  
语句已终止。

3.不违反约束  $sage \geq 0$  时可以看到插入成功:

```

insert into Worker values('00007','陈陈','1',30,'研发部')
select * from Worker

```

结果		消息			
	Number	Name	Sex	Sage	Department
1	00001	李勇	M	14	科技部
2	00002	王勇	M	38	科技部
3	00003	王浩	1	25	研发部
4	00004	陈	1	30	研发部
5	00007	陈陈	1	30	研发部

4.添加约束 U4 代码与结果如下，可以看到失败，因为与约束 U3 冲突（两个约束均针对 sage）。

```

alter table Worker
add constraint U4 check(sage<0)

```

消息

消息 547，级别 16，状态 0，第 1 行  
 ALTER TABLE 语句与 CHECK 约束"U4"冲突。该冲突发生于数据库"School"，表"dbo.Worker"，column 'Sage'。

5.加入规则的代码与结果如下：

```
go
create rule sage_range as @value between 1 and 100
go
exec sp_bindrule sage_range, 'worker.[sage]'
```



消息

已将规则绑定到表的列。

6. 插入数据失败，因为约束了  $1 \leq \text{sage} \leq 100$ ，显然  $\text{sage}=101$  不符合约束：

```
insert into Worker values('00009','王','F',101,'科技部')
```



消息

消息 513，级别 16，状态 0，第 1 行  
列的插入或更新与先前的 CREATE RULE 语句所指定的规则发生冲突。该语句已终止。冲突发生于数据库 'School'，表 'dbo.Worker'，列 'Sage'。  
语句已终止。

7. 解除绑定后显然可以插入成功，代码与结果如下：

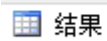
```
exec sp_unbindrule 'worker.[sage]'  
insert into Worker values ('00009','王','F',101,'科技部')
```



已解除了表列与规则之间的绑定。

(1 行受影响)

```
select * from Worker
```



	Number	Name	Sex	Sage	Department
1	00001	李勇	M	14	科技部
2	00002	王勇	M	38	科技部
3	00003	王浩	1	25	研发部
4	00004	陈	1	30	研发部
5	00007	陈陈	1	30	研发部
6	00009	王	F	101	科技部

8.添加规则 `sage>50` 发现可以添加成功 (即使前面有数据不满足 `sage > 50`) :

```
go
create rule rangesage as @value>50
go
exec sp_bindrule rangesage, 'worker.[sage]'
```



已将规则绑定到表的列。

## 五、实验体会：

通过这次实验，了解了如何自定义约束，实践了用户完整性，以及如何保护用户完整性；总体来说本次实验比较简单，完成的还算顺利。