

实验七：数据库的完整性

一、实验目的

- 理解并掌握关系数据库完整性的运行机制
 - 完整性约束定义>完整性约束检查>违约处理
- 理解并掌握关系数据库完整性主要约束类型及其含义和作用
 - PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, CHECK
- 理解并掌握关系数据库完整性定义、修改、删除和命名的方法
 - CREATE TABLE, ALTER TABLE
- 熟练掌握 Mysql8.4 下通过**系统表**查看完整性信息(PK, FK, UNIQUE, CHECK)的方法
 - INFORMATION_SCHEMA.TABLE_CONSTRAINTS 表
 - INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS 表
 - INFORMATION_SCHEMA.CHECK_CONSTRAINTS 表
 - INFORMATION_SCHEMA.KEY_COLUMN_USAGE 表
 - SHOW CREATE TABLE tbl_name;

二、实验要求

- 设计正确的 SQL 语句并测试其是否实现了完整性要求

- 完成实验内容，撰写并提交实验报告（按实验报告模板）
- 实验报告提交截止日期：2025 年 5 月 17 日 00:00（星期六）
- 选项：自行完成教材上的例题，但无需放入实验报告中

三、实验内容与步骤

本实验分成两部分：一是验证主码、外码、唯一约束和 check 约束的执行效果，其中包括给约束命名；二是介绍在 mysql8.4 下如何查询已定义的约束信息。

1.约束的定义与效果验证

(1) 创建两张表：雇员表 Emp 和工作表 Work，它们的表结构如下：

Emp 表			
字段	含义	数据类型	是否空
Eid	雇员编号	定长字符型，长度为 5	否
Ename	雇员姓名	变长字符型，长度为 10	/
WorkID	工作编号	定长字符，长度为 3	/
Salary	工资	数值型，总长度为 8，包括两位小数	/
Phone	电话号码	定长字符型，长度为 11	否

Work 表			
字段	含义	数据类型	是否空
WorkID	工作编号	定长字符，长度为 3	否
LowerSalary	最低工资	数值型，总长度为 8，包括两位小数	/
UpperSalary	最高工资	数值型，总长度为 8，包括两位小数	/

(2) 分别为两张表插入以下数据，观察插入操作是否成功。

emp 表数据：{('10001','Smith','001',2000,'13800010001'),
('10001', 'Jonny', '001', 3000, '13600010002'), ('10002', 'Mary', '002', 2500, '13800020002')}

work 表数据：{('001', 1000, 5000), ('002', 2000, 8000)}

(3) 设置 emp 表的 eid 为主码，观察该操作是否成功。若不成功，请分析原因并思考如何处理才能添加主码约束成功。添加主码约束命令：**alter table emp add primary key (eid);**

(4) 在成功设置 eid 为 emp 表的主码后，发布语句：

alter table emp add constraint pk_emp_eid primary key (eid);

观察执行结果并分析原因，记住系统的结果提示以便与下面的步骤（10）比较

(5) 分别使用以下两条命令查看主码约束信息，观察结果的异同：

show create table emp;

select * from information_schema.table_constraints where table_name='emp';

(6) 设置 emp 表的 workid 字段为外码，它引用 work 表中的 workid 字段，查看操作是否成功？若不成功说明原因，然后根据系统提示来修改 work 表的结构，使得 work 表满足 emp.workid 为外码的要求

(7) 第二次执行(6)，设置 emp 表的 workid 为外码，但不给出显式外码约束名，而由系统默认

(8) 再次执行步骤(7)

(9) 第三次执行步骤(6)，即设置 emp 表的 workid 为外码，但要求给出外码约束名，即 **fk_emp_work**

(10) 发布语句：**show create table emp;** 查看 emp 表的外码约束信息，理解系统默认的外码约束名的构成及自命名外码约束名，分析为什么外码约束名可以有多个，而主码约束只能有一个（对比步骤（4）的结果），总结 mysql 中主码约束命名与外码约束命名的规律

(11) 设置 work 表的 workid 为唯一值（唯一约束名由系统默认，无需自命名），连续两次执行该命令

(12) 发布语句：**show create table work;** 查看 work 表的唯一约束信息，理解系统默认唯一约束名的构成，总结 mysql 中唯一约束名的规律

(13) 发布语句：**alter table emp add constraint ck_salary check(3000<=salary);** 观察

操作是否成功？若不成功说明原因

(14) 修改 work 表以保证最低工资 lowersalary 一定不高于最高工资 uppersalary，要求给出约束名，即 ck_lower_upper_salary

(15) 给 work 表插入数据('003',4000,3000)，观察操作是否成功？若不成功说明原因

(16) 发布语句：**alter table work add check(lowersalary<=uppersalary);** 观察操作是否成功？若不成功说明原因

(17) 给 work 表插入数据('003',3000,4000)，观察操作是否成功？若不成功说明原因

(18) 再次发布语句：**alter table work add check(lowersalary<=uppersalary);** 观察操作是否成功？若不成功说明原因

(19) 发布语句：**show create table work;** 查看 work 表的 **check 约束** 信息，理解并总结系统默认的 **check 约束名** 的构成方法

(20) 以下为约束的级联操作验证（这部分在实验 5 中有要求，此处属于重新复习一遍）。删除 emp 表上所有的外码约束，重建外码约束，emp 表的 workid 字段引用 work 表的 workid 字段，要求外码约束中包含 **on delete cascade** 选项，外码约束名为 fk_emp_work

(21) 先查询 emp 表和 work 表中所有数据

(22) 发布语句：**delete from work where workid='001';**

(23) 发布语句：**select * from emp;** 观察执行结果并与 (21) 中查询 emp 表的结果进行比对，验证 **on delete cascade** 选项是否发生作用

2.约束查询

mysql8 中定义的约束信息主要放在 **INFORMATION_SCHEMA** 数据库的以下表中：

- **TABLE_CONSTRAINTS** 表（主码、外码、唯一约束、check 约束均可查询该表得到，如果存在）
- **REFERENTIAL_CONSTRAINTS** 表（提供外码引用的信息，包括删除规则，更新规则）
- **CHECK_CONSTRAINTS** 表（提供 check 所在的库、check 约束名及 check 语句内容等信息）
- **KEY_COLUMN_USAGE** 表（提供**键列**上约束的详细信息，包括主码、外码）

以上表结构都可以使用 **desc tbl_name;**命令得到，如 desc table_constraints;

如果要获得某张表上**约束的定义语句**，可使用以下命令获得：

- **SHOW CREATE TABLE tbl_name;** -- 将 *tbl_name* 替换为实际的表

```
mysql> show create table employees;
+-----+-----+
| Table | Create Table |
+-----+-----+
| employees | CREATE TABLE `employees` (
  `employee_id` smallint NOT NULL,
  `first_name` varchar(255) NOT NULL,
  `last_name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `phone` varchar(50) NOT NULL,
  `hire_date` date NOT NULL,
  `manager_id` smallint DEFAULT NULL,
  `job_title` varchar(255) NOT NULL,
  PRIMARY KEY (`employee_id`),
  KEY `fk_employees_manager` (`manager_id`),
  CONSTRAINT `fk_employees_manager` FOREIGN KEY (`manager_id`) REFERENCES `employees` (`employee_id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)
```

查询上述 4 张表并验证表中信息是否为第一部分（**1.约束的定义与效果验证**）操作中的主码、外码、唯一约束和 check 约束信息

3.删除表上的所有约束

- (1) 删除 emp 表上的所有约束（包括主码、外码、唯一约束和 check 约束）
- (2) 删除 work 表上的所有约束（包括主码、外码、唯一约束和 check 约束）

四、实验思考

- 1.总结 mysql8.4 中以下约束名的规律：主码、外码、唯一约束和 check 约束
- 2.对于一个相同的约束，如 alter table work unique(workid);可能有多个不同的约束名，请问这些不同的约束名对应不同的约束效果还是同一个约束效果？即，具有不同约束名的同一约束的效果是否不同？

注：将问题的解答放到实验报告的实验总结部分

五、补充资料

- 1.MySQL 之约束（主键、唯一、非空、自增、外键）：

<https://www.cnblogs.com/wenyuan519/p/18207770>

对外键约束的说明是亮点

“外键约束不是必须的，而且现在很多大的公司，数据量比较大时，不建议在数据库层面设计外键，因为他觉得这样效率低，把这个数据的约束挪到代码层面去判断。”

2. MySQL 表的约束

https://blog.csdn.net/2203_76003626/article/details/144311908

- 3.mysql 如何设置约束名

https://blog.51cto.com/u_16213408/9238792