实验七：数据库的完整性

1. 实验环境

- 华为 ECS+openGauss 数据库服务器平台
- **前提：**openGauss 数据库服务器正常运行

2. 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、删除和重命名的方法
  - CREATE TABLE，ALTER TABLE
- 熟练掌握 openGauss 下通过系统表 pg_constraint 查看完整性信息的方法
- 熟练掌握 openGauss 下通过查看表结构来查看主外码信息的方法
- 熟练掌握 openGauss 下通过查看完整性约束定义的方法

3. 实验要求

- 完成实验内容并提交实验报告到 FTP 上的相应文件夹“实验七”。
- 实验报告提交截至日期：2024 年 5 月 18 日。
- 请自行完成教材上的相关例题，但无需将其放到本实验报告中

4. 实验内容与步骤

（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）分别为两张表插入如下数据，查看插入操作是否成功。

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

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

（3）修改雇员表的结构，设置 Eid 为主码，主码名称为 eid_pk，查看该操作是否成功。若不成功，请说明原因并思考如何处理才能成功添加约束。要求：所有约束都要显式给出约束名，不可由系统默认，因为删除约束时需要用到约束名。

（4）将 eid 为主码的约束名 eid_pk 改为 pk_eid。

（5）设置雇员表中的 phone 字段取唯一值，查看该操作是否成功？若不成功说明原因。

（6）给雇员表添加一条新记录('10003','Amy','002', 3000,'13800020003')，查看执行结果。

（7）设置工作表的 WorkID 为主码。

（8）修改雇员表，设置雇员表的 WorkID 字段为外码，它引用工作表中的 WorkID 字段，查看操作是否成功？若不成功说明原因。

（9）给雇员表添加一条新记录('10003','Amy', '003', 3000, '13800020003')，查看操作是否成功？若不成功说明原因。

（10）在雇员表中，设置雇员工资必须大于或等于 1000。查看操作是否成功？若不成功说明原因。

（11）给雇员表添加一条新记录('10003','Robert','002',500,'13800020003')，查看执行操作是否成功？若不成功说明原因。

（12）在工作表中，设置其最低工资不超过最高工资。

（13）给工作表添加一条新记录('002',4000,3000)，查看操作是否成功？若不成功说明原因。

（14）通过查看 openGauss 的系统表 pg_constraint 了解表上的约束。

（15）通过 gsql 命令\d+ table_name 查看该表上的约束定义。

（16）删除雇员表的所有约束，包括主码约束、外码约束和其他约束。

（17）删除工作表所有约束，包括主码约束。

5. 实验思考

- openGauss 实现完整性规则的机制是什么？在 SQL 语句中实现完整性规则的常见约束有哪些？各自适用什么业务场景？

6. 语法参考

- 创建表语法：

```
CREATE TABLE <表名>
(<列名><数据类型> DEFAULT <缺省值>] [列级约束定义],
<列名><数据类型> DEFAULT <缺省值>] [列级约束定义],
...,
[<表级约束定义>, …,<表级约束定义>]);
```

列级约束定义形式： [CONSTRAINT <约束名>]<列约束>

– 常用列级约束：NOT NULL, PRIMARY KEY, UNIQUE, CHECK(<条件>)

行级约束定义形式： [CONSTRAINT <约束名>]<表约束>

– 常用表级约束：PRIMARY KEY, UNIQUE, CHECK(<条件>), REFERENCES (A1, …, Ak) REFERENCES <外表名> (<外表主码>) [<参照触发动作>],＜参照触发动作＞说明违反参照完整性时需要采取的措施

- 修改表上约束语法：

```
ALTER TABLE <表名>
[ADD <表约束定义>]
[DROP CONSTRAINT <约束名>{CASCADE | RESTRICT}]
[RENAME CONSTRAINT constraint_name to new_constraint_name];
```

<表约束定义>的两种形式

– 不带约束名：ADD 约束定义，如，ADD UNIQIE(EID);

– 带约束名：ADD CONSTRAINT constraint_name;

– **注意：**带约束名必须 CONSTRAINT 和 constraint_name 同时出现，否则报错，如 ADD CONSTRAINT PRIMARY KEY(EID)是错误的。

```
DROP CONSTRAINT <约束名>{CASCADE | RESTRICT}; --实现删除已创建的约束。
RENAME CONSTRAINT constraint_name to new_constraint_name; --实现对已有约束的改名
```

- 查看表上创建的约束(类型和定义)：

– 法 1：\d+ table_name，如，\d+ emp

– 法 2：查询系统表 pg_constraint。pg_constraint 存储 check 约束、主码约束和唯一值约束。

步骤：

（1）select oid, conname from pg_constraint;来查看约束名 conname 的 oid，**注意：**oid 是隐式的，通过命令\d pg_constraint 是看不到该字段的，必须显式加上。

（2）利用查询到的 oid 使用 pg_get_constraintdef(constraint_oid)函数查看约束源码：

```
select  pg_get_constraintdef(constraint_oid);
```