

数据库系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 数据库的完整性 |
| 实验日期： | 2022/5/5 |
| 实验地点： | 四号楼 |
| 提交日期： | 2022/5/6 |
|  | |
| 学号： | 22920202202879 |
| 姓名： | 陈奕培 |
| 专业年级： | 软工2020级 |
| 学年学期： | 2021-2022学年第二学期 |

1. 实验目的

* 掌握数据库的特点（字体：华文仿宋，字号：四号，下同）

1. 实验内容和步骤

（1）创建两张表

CREATE TABLE Emp(

    Eid CHAR(5) NOT NULL,

    Ename VARCHAR(10),

    WorkID CHAR(3),

    Salary NUMBER(8,2),

    Phone CHAR(11) NOT NULL

);

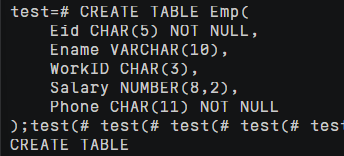
CREATE TABLE Work(

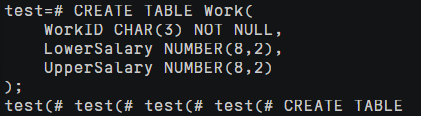
    WorkID CHAR(3) NOT NULL,

    LowerSalary NUMBER(8,2),

    UpperSalary NUMBER(8,2)

);





（2）插入数据

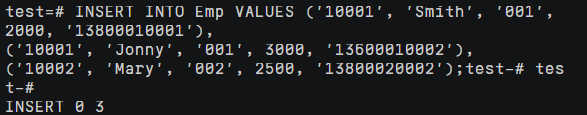
INSERT INTO Emp VALUES ('10001', 'Smith', '001', 2000, '13800010001'),

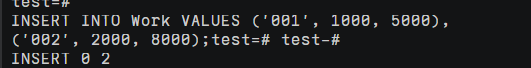
('10001', 'Jonny', '001', 3000, '13600010002'),

('10002', 'Mary', '002', 2500, '13800020002');

INSERT INTO Work VALUES ('001', 1000, 5000),

('002', 2000, 8000);

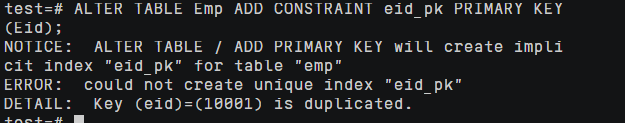




（3）为Emp添加主键

ALTER TABLE Emp ADD CONSTRAINT eid\_pk PRIMARY KEY(Eid);

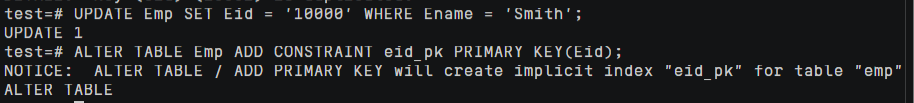
添加失败



原因是eid=10001重复出现了两次，违反了实体完整性约束

将其中一个eid改为10000即可，例如

UPDATE Emp SET Eid = '10000' WHERE Ename = 'Smith';



（4）修改约束名

ALTER TABLE Emp RENAME CONSTRAINT eid\_pk TO pk\_eid;



（5）设phone为唯一值

ALTER TABLE Emp ADD CONSTRAINT uni\_phone UNIQUE(Phone);



（6）给雇员表添加一条新记录(‘10003’,’Amy’,’002’, 3000,’13800020003’)

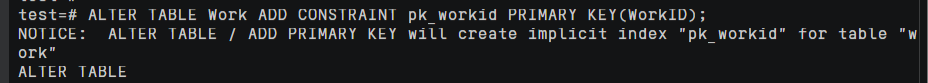
INSERT INTO Emp VALUES ('10003', 'Amy', '002', 3000, '13800020003');



插入成功，因为id不为空且不重复，phone也不重复

（7）为Work设置主键

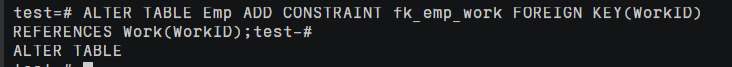
ALTER TABLE Work ADD CONSTRAINT pk\_workid PRIMARY KEY(WorkID);



（8）为Emp设置外键

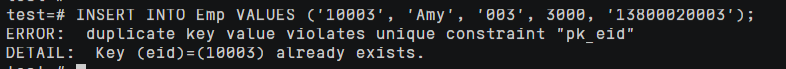
ALTER TABLE Emp ADD CONSTRAINT fk\_emp\_work FOREIGN KEY(WorkID)

REFERENCES Work(WorkID);



（9）给雇员表添加一条新记录(‘10003’,’Amy’, ‘003’, 3000, ‘13800020003’)

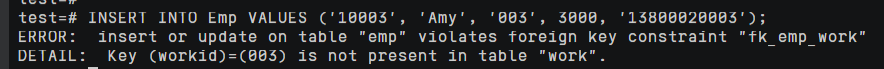
INSERT INTO Emp VALUES ('10003', 'Amy', '003', 3000, '13800020003');



插入失败，因为Emp中存在eid为10003的元组



删除之前的元组后



任然失败，原因是work表中不存在workid=003的元组，添加的数据不满足参照完整性定义约束

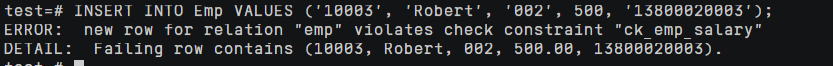
（10）在雇员表中，设置雇员工资必须大于或等于 1000

ALTER TABLE Emp ADD CONSTRAINT ck\_emp\_salary CHECK(Salary >= 1000);



（11）给雇员表添加一条新记录(‘10003’,’Robert’,‘002’,500,‘13800020003’)

INSERT INTO Emp VALUES ('10003', 'Robert', '002', 500, '13800020003');



插入失败，违反了用户定义的完整性约束

（12）在工作表中，设置其最低工资不超过最高工资

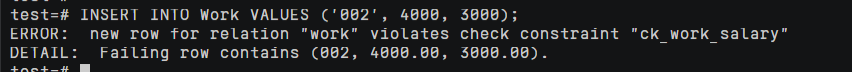
ALTER TABLE Work ADD CONSTRAINT ck\_work\_salary

CHECK(UpperSalary >= LowerSalary);



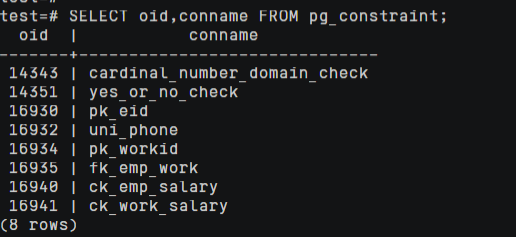
（13）给工作表添加一条新记录(‘002’,4000,3000)

INSERT INTO Work VALUES ('002', 4000, 3000);

插入失败，违反了用户定义的完整性约束

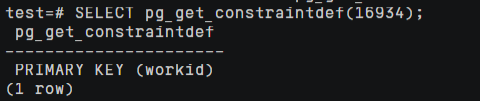
（14）通过查看 openGauss 的系统表 pg\_constraints 了解表上的约束

SELECT oid,conname FROM pg\_constraint;



先得到约束的oid

SELECT pg\_get\_constraintdef(16934);

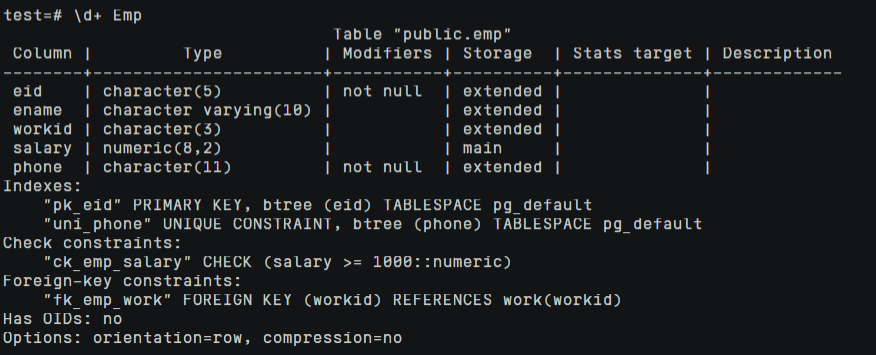


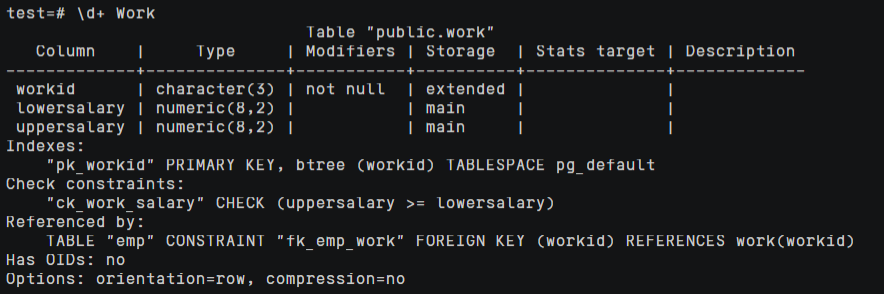
再通过oid去查看约束的详情

（15）通过 gsql 命令\d+ table\_name 查看改表上的约束定义

\d+ Emp

\d+ Work



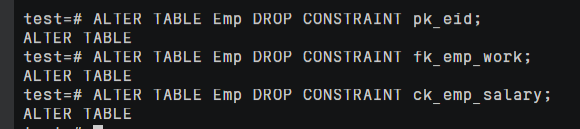


（16）删除雇员表的所有约束，包括主码约束、外码约束和其他约束

ALTER TABLE Emp DROP CONSTRAINT pk\_eid;

ALTER TABLE Emp DROP CONSTRAINT fk\_emp\_work;

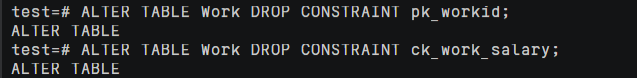
ALTER TABLE Emp DROP CONSTRAINT ck\_emp\_salary;



（17）删除工作表所有约束，包括主码约束

ALTER TABLE Work DROP CONSTRAINT pk\_workid;

ALTER TABLE Work DROP CONSTRAINT ck\_work\_salary;



实验思考

**·openGauss实现完整性规则的机制是什么？**

通过设置完整性约束来是实现完整性约束；

**·在SQL语句中实现完整性规则的常见约束有哪些？各自适应什么业务场景？**

有三种常见约束：实体完整性约束，参照完整性约束，用户定义的完整性约束；

实体完整性适用于需要主键，或某值需取唯一值时；

参照完整性适用于某值需要参照另外一个表中的主码的时候，通常用于表达表之间的关系；

用户定义的完整性约束适用于用户的特定需求；

1. 实验总结

3.1 完成的工作

创建两张表；

为表插入数据；

为表添加主键；

修改主键的约束名；

为表添加unique约束；

为表添加外键；

为表定义check语句；

删除表的约束；

3.2 对实验的认识

掌握了表的约束的添加方法；

修改表的约束名的方法；

当不满足完整性约束时，数据库会拒绝操作；

删除表的约束的方法；

3.3 遇到的困难及解决方法

无