**暨南大学本科实验报告专用纸**

课程名称 《数据库系统》 成绩评定

实验项目名称 指导教师雷小林 、魏林锋

实验项目编号 实验项目类型 综合 实验地点 N117

学生姓名 金鸿正 学号 2021102560

学院 信息科学与技术学院 系 计算机 专业 软件工程

实验时间 年 月 日 午～ 月 日 午 温度 ℃湿度

3 完整性语言实验

3.1 实体完整性实验

3.1.1 实验目的

掌握实体完整性的定义和维护方法。

3.1.2 实验内容和要求

定义实体完整性，删除实体完整性。能够写出两种方式定义实体完整性的 SQL 语句：创建表时定义实体完整性、创建表后定义实体完整性。设计 SQL 语句验证完整性约束是否起作用。

3.1.3 实验重点和难点

实验重点：创建表时定义实体完整性。

实验难点：有多个候选码时实体完整性的定义。

3.1.4 实验内容记录

这里参照 employees 数据库创建一个新的数据库 test;

CREATE DATABASE test;

3.1.4.1 创建表时定义实体完整性（列级实体完整性）

创建雇员表。

CREATE TABLE employees (

emp\_no INT NOT NULL PRIMARY KEY,

birth\_date DATE NOT NULL,

first\_name VARCHAR(14) NOT NULL,

last\_name VARCHAR(16) NOT NULL,

gender ENUM ('M','F') NOT NULL,

hire\_date DATE NOT NULL

);

3.1.4.2 创建表时定义实体完整性（表级实体完整性）

创建雇员表。

CREATE TABLE employees (

emp\_no INT NOT NULL,

birth\_date DATE NOT NULL,

first\_name VARCHAR(14) NOT NULL,

last\_name VARCHAR(16) NOT NULL,

gender ENUM ('M','F') NOT NULL,

hire\_date DATE NOT NULL,

CONSTRAINT PK\_empno PRIMARY KEY(emp\_no)

);

注：MYSQL 不支持约束命名。

3.1.4.3 创建表后定义实体完整性

创建雇员表。

CREATE TABLE employees (

emp\_no INT NOT NULL,

birth\_date DATE NOT NULL,

first\_name VARCHAR(14) NOT NULL,

last\_name VARCHAR(16) NOT NULL,

gender ENUM ('M','F') NOT NULL,

hire\_date DATE NOT NULL

);

ALTER TABLE employees

ADD PRIMARY KEY(emp\_no);

3.1.4.4 定义实体完整性（多属性主码）

创建职位表。

CREATE TABLE titles (

emp\_no INT NOT NULL,

title VARCHAR(50) NOT NULL,

from\_date DATE NOT NULL,

to\_date DATE,

FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no) ON DELETE CASCADE,

PRIMARY KEY (emp\_no,title, from\_date)

);

3.1.4.5 有多个候选码时定义实体完整性

当存在多个候选码时，只能定义一个主码，其它的候选码定义唯一性约束。

创建部门表。

CREATE TABLE departments (

dept\_no CHAR(4) NOT NULL,

dept\_name VARCHAR(40) NOT NULL,

PRIMARY KEY (dept\_no),

UNIQUE KEY (dept\_name)

);

3.1.4.6 删除实体完整性

删除部门表上的实体完整性。

ALTER TABLE departments

DROP INDEX dept\_name;

ALTER TABLE departments

DROP PRIMARY KEY;

3.1.5 思考

所有列级完整性约束都可以改写为表级完整性约束，而表级完整性约束不一定能改写为列级完整性约束。请举例说明。

例如多属性主码就只能通过表级完整性约束定义。

什么情况下会违反实体完整性约束，DBMS 将做何种违约处理？请用实验验证。

当更新数据时，新的数据可能违反实体完整性约束。此时 DBMS 会拒绝执行。

举例：略。

3.2 参照完整性实验

3.2.1 实验目的

掌握参照完整性的定义和维护方法。

3.2.2 实验内容和要求

定义参照完整性，定义参照完整性的违约处理，删除参照完整性。写出两种方式定义参照完整性的 SQL 语句：创建表时定义参照完整性、创建表后定义参照完整性。

3.2.3 实验重点和难点

实验重点：创建表时定义参照完整性。

实验难点：参照完整性的违约处理定义。

3.2.4 实验内容记录

3.2.4.1 创建表时定义参照完整性

在已有员工表的情况下，定义工资表。

列级参照完整性定义。

CREATE TABLE salaries (

emp\_no INT NOT NULL REFERENCES employees (emp\_no),

salary INT NOT NULL,

from\_date DATE NOT NULL,

to\_date DATE NOT NULL,

PRIMARY KEY (emp\_no, from\_date)

);

注：经测试，MYSQL 定义列级外键约束无效。

表级参照完整性定义。

注：MYSQL 支持外键约束命名。

CREATE TABLE salaries (

emp\_no INT NOT NULL,

salary INT NOT NULL,

from\_date DATE NOT NULL,

to\_date DATE NOT NULL,

PRIMARY KEY (emp\_no, from\_date),

CONSTRAINT FK\_empno FOREIGN KEY (emp\_no) REFERENCES employees(emp\_no)

);

3.2.4.2 创建表后定义参照完整性

CREATE TABLE salaries (

emp\_no INT NOT NULL,

salary INT NOT NULL,

from\_date DATE NOT NULL,

to\_date DATE NOT NULL,

PRIMARY KEY (emp\_no, from\_date)

);

ALTER TABLE salaries

ADD CONSTRAINT FK\_empno

FOREIGN KEY(emp\_no) REFERENCES employees(emp\_no);

3.2.4.3 定义参照完整性的违约处理

定义两张表：

CREATE TABLE departments (

dept\_no CHAR(5) NOT NULL,

dept\_name VARCHAR(40) NOT NULL,

PRIMARY KEY (dept\_no)

);

CREATE TABLE emp\_dept (

emp\_no CHAR(10) NOT NULL,

dept\_no CHAR(5),

CONSTRAINT FK\_deptno

FOREIGN KEY (dept\_no)

REFERENCES departments(dept\_no)

ON DELETE SET NULL

ON UPDATE SET NULL,

PRIMARY KEY(emp\_no)

);

注：如果要定义删除后的行为，则必须不能与已有的限制冲突（如不能定义 NOT NULL）。

插入两条数据：

INSERT INTO departments

VALUES('00001', 'Finance');

INSERT INTO emp\_dept

VALUES('10000', '00001');

然后删除被参照表中的元组，之后参照表会变成：

+--------+---------+

| emp\_no | dept\_no |

+--------+---------+

| 10000 | NULL |

+--------+---------+

3.2.4.4 删除参照完整性

删除 emp\_dept 上的外码。

注：和书中写法稍有不同。

ALTER TABLE emp\_dept

DROP FOREIGN KEY FK\_deptno;

3.2.5 思考

对于自引用表，例如课程表（课程号、课程名、先修课程号、学分）中的先修课程号引用该表的课程号，请完成如下任务：

（1）写出课程表上的实体完整性和参照完整性。

（2）在考虑实体完整性约束的条件下，试举出几种录入课程数据的方法。

CREATE TABLE course (

cno CHAR(4),

cname VARCHAR(40),

cpno CHAR(4),

ccredit SMALLINT

);

ALTER TABLE course

ADD PRIMARY KEY(cno);

ALTER TABLE course

ADD CONSTRAINT FK\_cpno

FOREIGN KEY(cpno) REFERENCES course(cno);

在录入数据时，可以按照引用的顺序录入数据（总是先录入不存在引用或引用已存在的数据），也可以临时移除完整性约束，在录入数据后，再添加完整性约束。

如果数据本身无法满足约束，则引入约束时会失败。

比如先加入数据：

INSERT INTO course

VALUES('0001','DataStructure','0002',80);

这一条数据存在依赖问题，此时再引入完整性约束，则会出现无法引入的情况。

3.3 用户自定义完整性实验

3.3.1 实验目的

掌握用户自定义完整性的定义和维护方法。

3.3.2 实验内容和要求

针对具体应用语义，选择 NULL、NOT NULL、DEFAULT、DEFAULT、UNIQUE、CHECK 等，定义属性上的约束条件。

3.3.3 实验重点和难点

实验重点：NULL、NOT NULL、DEFAULT

实验难点：CHECK

3.3.4 实验内容记录

3.3.4.1 定义属性 NULL、NOT NULL 约束

注：默认约束即为 NULL 约束，即可以为 NULL 。

这里不作示例。

3.3.4.2 定义属性 DEFAULT 约束

DEFAULT 约束用于给定属性的默认值，即不提供值的时候自动填充的值。

这里不作示例。

3.3.4.3 定义 UNIQUE 约束

UNIQUE 约束即必须唯一。

3.3.4.4 使用 CHECK 约束条件

CHECK 可以跟表达式，并且可以引用多个属性。

如：

...

CHECK (from\_date <= to\_date),

...

CHECK (grade >= 0 AND grade <=100)

...

**暨南大学本科实验报告专用纸(附页)**