

数据库实验2

赵丰 091242045

一. 实验要求

利用DB2，创建一个新的数据库，在该数据库下创建如下表格：

职工（姓名，工号，出生日期，家庭地址，年薪，管理员工号，所在部门编号）；

部门（部门名称，部门编号，部门负责人的工号）；

项目（项目名称，项目编号，所在城市，主管部门编号）；

工作（职工工号，项目编号，工作时间）；

家属（职工工号，家属的姓名，家属的性别）。

要求：

1. 表名和属性名用英文字符表示；
2. 表中各属性类型自行设计；
3. 分析该关系模式中存在的各种数据约束，并在创建表时体现出来；
4. 加入样本数据（自行设计），执行违反数据约束的操作，并观察结果；
5. 创建一个触发器，当职工参加一个新的项目时，年薪增加2%；
6. 创建一个用户，将工作表的查询权限赋予该用户。

二. 实验过程

1. 创建数据库company

----- 输入的命令 -----

```
create database company;
```

```
create database company
```

DB20000I CREATE DATABASE 命令成功完成。

所有数据库							
名称	系统	实例	类型	数据库	路径	注释	
 COMPANY	ZHAOFENG_PC	DB2	本地	COMPANY	D:		
 TOOLSDB	ZHAOFENG_PC	DB2	本地	TOOLSDB	D:		

2. 连接数据库

----- 输入的命令 -----

```
connect to company;
```

```
connect to company
```

数据库连接信息

数据库服务器 = DB2/NT 9.7.3

SQL 授权标识 = ZHAOFENG

本地数据库别名 = COMPANY

与目标的 JDBC 连接已成功。

3. 创建表employee

----- 输入的命令 -----

```
create table employee(  
    ename char(30) not null,  
    enumber int not null primary key,  
    ebirthday char(30) not null,  
    eaddress char(30),  
    esalary int,  
    mnumber int,  
    dnumber int);
```

```
create table employee( ename char(30) not null, enumber int not null primary key,  
ebirthday char(30) not null, eaddress char(30), esalary int, mnumber int, dnumber  
int)
```

DB20000I SQL 命令成功完成。

4. 创建表department

----- 输入的命令 -----

```
create table department(  
    dname char(30),  
    dnumber int not null primary key,  
    enumber int);
```

```
create table department( dname char(30), dnumber int not null primary key, enumber  
int)
```

DB20000I SQL 命令成功完成。

5. 创建表project

----- 输入的命令 -----

```
create table project(  
    pname char(30),  
    pnumber int not null primary key,  
    pcity char(30),  
    dnumber int);
```

```
create table project( pname char(30), pnumber int not null primary key, pcity  
char(30), dnumber int)
```

DB20000I SQL 命令成功完成。

6. 创建表work

----- 输入的命令 -----

```
create table work(  
    enumber int,  
    pnumber int,  
    wtime int);
```

```
create table work( enumber int, pnumber int, wtime int)
```

DB20000I SQL 命令成功完成。

7. 创建表relation

----- 输入的命令 -----

```
create table relation(  
    enumber int,  
    rname char(30),  
    rsex char(1));
```

```
create table relation( enumber int, rname char(30), rsex char(1))
```

DB20000I SQL 命令成功完成。

ZHAOFENG_PC - DB2 - COMPANY - 表									
名称	模式	表空间	注释	索引表空间	大数据表空间	类型	基数	统计信息时间	
DEPARTMENT	ZHAOFENG	USERSPACE1				I	-1		
EMPLOYEE	ZHAOFENG	USERSPACE1				I	-1		
PROJECT	ZHAOFENG	USERSPACE1				I	-1		
RELATION	ZHAOFENG	USERSPACE1				I	-1		
WORK	ZHAOFENG	USERSPACE1				I	-1		

8. 给表project添加数据约束

----- 输入的命令 -----

```
alter table project add  
    constraint fk_project_dnumber  
    foreign key(dnumber)  
    references department(dnumber)  
        on update restrict  
        on delete restrict;
```

```
alter table project add constraint fk_project_dnumber foreign key(dnumber)  
references department(dnumber) on update restrict on delete restrict
```

DB20000I SQL 命令成功完成。

9. 给表work添加数据约束

----- 输入的命令 -----

```
alter table work add
```

```

        constraint fk_work_enumber
        foreign key(enumber)
        references employee(enumber)
        on update restrict
        on delete restrict;
-----
alter table work add constraint fk_work_enumber foreign key(enumber) references
employee(enumber) on update restrict on delete restrict
DB20000I  SQL 命令成功完成。

```

```

----- 输入的命令 -----
alter table work add
        constraint fk_work_pnumber
        foreign key(pnumber)
        references project(pnumber)
        on update restrict
        on delete restrict;
-----
alter table work add constraint fk_work_pnumber foreign key(pnumber) references
project(pnumber) on update restrict on delete restrict
DB20000I  SQL 命令成功完成。

```

10. 给表relation添加数据约束

```

----- 输入的命令 -----
alter table relation add
        constraint fk_relation_enumber
        foreign key(enumber)
        references employee(enumber)
        on update restrict
        on delete restrict;
-----
alter table relation add constraint fk_relation_enumber foreign key(enumber)
references employee(enumber) on update restrict on delete restrict
DB20000I  SQL 命令成功完成。

```

11. 给表employee添加数据

```

----- 输入的命令 -----
insert into employee values('张三', 101, '1981/1/1', '北京', 10000, null, 201);
insert into employee values('李四', 102, '1982/2/2', '上海', 20000, 101, 201);
insert into employee values('王五', 103, '1983/3/3', '南京', 30000, null, 202);
-----
insert into employee values('张三', 101, '1981/1/1', '北京', 10000, null, 201)
DB20000I  SQL 命令成功完成。
insert into employee values('李四', 102, '1982/2/2', '上海', 20000, 101, 201)

```

DB20000I SQL 命令成功完成。

```
insert into employee values('王五', 103, '1983/3/3', '南京', 30000, null, 202)
```

DB20000I SQL 命令成功完成。

ZHAOFENG_PC - DB2 - COMPANY - ZHAOFENG.EMPLOYEE							
采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”笔记本来更改编辑格式。							
ENAME	ENUMBER	EBIRTHDAY	EADDRESS	ESALARY	MINUMBER	DIUMBER	
张三 ...	101	1981/1/1 ...	北京 ...	10000		201	
李四 ...	102	1982/2/2 ...	上海 ...	20000	101	201	
王五 ...	103	1983/3/3 ...	南京 ...	30000		202	

12. 给表department添加数据

----- 输入的命令 -----

```
insert into department values('执行', 201, 101);
```

```
insert into department values('宣传', 202, 103);
```

```
insert into department values('执行', 201, 101)
```

DB20000I SQL 命令成功完成。

```
insert into department values('宣传', 202, 103)
```

DB20000I SQL 命令成功完成。

ZHAOFENG_PC - DB2 - COMPANY - ZHAOFENG.DEPARTMENT			
采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”			
DNAME	DNUMBER	ENUMBER	
执行 ...	201	101	
宣传 ...	202	103	

13. 给表project添加数据

----- 输入的命令 -----

```
insert into project values('建造写字楼', 301, '北京', 201);
```

```
insert into project values('建造写字楼', 301, '北京', 201)
```

DB20000I SQL 命令成功完成。

ZHAOFENG_PC - DB2 - COMPANY - ZHAOFENG.PROJECT			
采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”笔记本来			
PNAME	PNUMBER	PCITY	DIUMBER
建造写字楼 ...	301	北京 ...	201

14. 给表work添加数据

----- 输入的命令 -----

```
insert into work values(101,301,100);
```

```
insert into work values(102,301,200);
```

```
insert into work values(101,301,100)
```

DB20000I SQL 命令成功完成。

```
insert into work values(102,301,200)
```

DB20000I SQL 命令成功完成。

ZHAOFENG_PC - DB2 - COMPANY - ZHAOFENG.WORK		
采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”		
ENUMBER	PNUMBER	WTIME
101	301	100
102	301	200

15. 给表relation添加数据

----- 输入的命令 -----

```
insert into relation values(101,'赵六','m');
```

```
insert into relation values(101,'赵六','m')
```

DB20000I SQL 命令成功完成。

ZHAOFENG_PC - DB2 - COMPANY - ZHAOFENG.RELATION		
采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”		
ENUMBER	RNAME	RSEX
101	赵六	m

16. 执行违反主键约束的操作

----- 输入的命令 -----

```
insert into employee values('钱七',101,'1984/4/4','香港',40000,101,201);
```

```
insert into employee values('钱七',101,'1984/4/4','香港',40000,101,201)
```

DB21034E 该命令被当作 SQL 语句来处理，因为它是无效的“命令行处理器”命令。在 SQL 处理期间，它返回：

SQL0803N INSERT 语句、UPDATE 语句或由 DELETE

语句导致的外键更新中的一个或多个值无效，因为由“1”

标识的主键、唯一约束或者唯一索引将表“ZHAOFENG.EMPLOYEE”

的索引键限制为不能具有重复值。 SQLSTATE=23505

SQL0803N INSERT 语句、UPDATE 语句或由 DELETE 语句导致的外键更新中的一个或多个值无效，因为由“1”标识的主键、唯一约束或者唯一索引将表“ZHAOFENG.EMPLOYEE

”的索引键限制为不能具有重复值。

说明：

INSERT 或 UPDATE 对象表 “<表名>” 被一个或多个 UNIQUE 索引约束为在某些列或列组中具有唯一值。另外，父表上的 DELETE 语句导致更新从属表 “<表名>”（该从属表受一个或多个 UNIQUE 索引约束）中的外键。唯一索引可能支持在表上定义的主键或唯一约束。不能处理语句，因为完成所请求的 INSERT、UPDATE 或 DELETE 语句将导致重复的列值。如果索引在 XML 列上，那么可以从单个 XML 文档中生成索引键的重复值。

另外，如果视图是 INSERT 或 UPDATE 语句的对象，那么在其上定义视图的表 “<表名>” 会受到约束。

如果 “<索引标识>” 为整数值，那么可以通过发出下列查询来从 SYSCAT.INDEXES 中获取索引名：

```
SELECT INDNAME, INDSHEMA
FROM SYSCAT.INDEXES
WHERE IID = <index-id>
AND TABSCHEMA = 'schema'
AND TABNAME = 'table'
```

其中，schema 表示 “<表名>” 的模式部分，table 表示 “<表名>” 的表名部分。

无法处理该语句。未更改表。

用户响应：

检查 “<索引标识>” 标识的索引的定义。

对于 UPDATE 语句，确保指定的操作本身与唯一性约束一致。如果这样做未发生错误，那么检查对象表内容以确定问题的原因。

对于 INSERT 语句，检查对象表内容以确定在指定的值列表中哪个值违反了唯一性约束。另一方面，如果 INSERT 语句包含了子查询，那么该子查询寻址的对象表内容必须与对象表内容相匹配以确定问题的原因。注意，如果该表受安全策略保护，那么表示您的 LBAC 凭证可能不允许您查看导致错误的行。

如果索引在 XML 列上并且语句为 INSERT 或 UPDATE，那么应考虑 XML 文档是否将导致单个 XML 文档中产生重复值。

对于 DELETE 语句，检查用规则 ON DELETE SET NULL 定义的外键上唯一约束的已标识从属表。此表具有外键列，该外键列包括在不能设置为 NULL（因为该表的

列中已存在 NULL) 的已标识唯一索引中。

联合系统用户：找出使请求失败的数据源中的问题（请参阅《故障诊断指南》以确定哪个数据源处理 SQL 语句失败），并对先前列示的条件检查索引定义和数据。

sqlcode: -803

sqlstate: 23505

17. 执行违反不允许空值约束的操作

----- 输入的命令 -----

```
insert into department values('销售',null,null);
```

```
insert into department values('销售',null,null)
```

DB21034E 该命令被当作 SQL 语句来处理，因为它是无效的“命令行处理器”命令。在 SQL 处理期间，它返回：

SQL0407N 不允许对 NOT NULL 列 "TBSPACEID=2, TABLEID=5, COLNO=1" 赋予空值。

SQLSTATE=23502

SQL0407N 不允许对 NOT NULL 列 "TBSPACEID=2, TABLEID=5, COLNO=1" 赋予空值。

说明：

发生了下列情况之一：

- * 更新或插入值为 NULL，但是对象列在表定义中已声明为 NOT NULL。因此：

- * 空值不可插入到该列中。

- * 更新不能将该列中的值设置为 NULL。

- * 触发器中的 SET 转换变量语句不能将该列中的值设置为 NULL。

- * 更新或者插入值是 DEFAULT，但是对象列被声明为 NOT NULL，而表定义中又没有 WITH DEFAULT。因此：

- * 缺省值 NULL 不能插入该列。

- * 更新不能在该列设置缺省值 NULL。

- * 触发器中的 SET 转换变量语句不能在该列中设置缺省值 NULL。

- * INSERT 语句的列名列表省略了一列，该列声明为 NOT NULL，而表定义中没有 WITH DEFAULT。

- * INSERT 语句的视图省略了一列，该列声明为 NOT NULL 而基本表定义中没有 WITH DEFAULT。

- * ALTER TABLE 语句试图使用 SET NOT NULL 来改变列，但是现有表数据包含一个具有空值的行。

如果“<名称>”的值的格式为“TBSPACEID=n1, TABLEID=n2, COLNO=n3”，那么当发生错误时，SQL 语句的列名不可用。所提供的值标识不允许空值的基本表的表空间、表和列号。

联合系统用户：联合服务器或数据源可以检测到此情况。某些数据源未向“<名称>”提供适当的值。在这些情况下，消息标记将具有以下格式：“<数据源>:未知”，指示指定的数据源的实际值未知。

无法处理该语句。

注：在某些情况下，标记“<名称>”可能不填充（SQLCA 的 sqlerrmc 字段不填充）。

用户响应：

检查对象表定义以确定该表的哪些列有 NOT NULL 属性且没有 WITH DEFAULT 属性后，更正该 SQL 语句。

“<名称>”的值的格式为“TBSPACEID=n1, TABLEID=n2, COLNO=n3”，那么可以使用下列查询来确定表名和列名：

```
SELECT C. TABSCHEMA, C. TABNAME,
       C. COLNAME
FROM SYSCAT. TABLES AS T,
     SYSCAT. COLUMNS AS C
WHERE T. TBSPACEID = n1
      AND T. TABLEID = n2
      AND C. COLNO = n3
      AND C. TABSCHEMA = T. TABSCHEMA
      AND C. TABNAME = T. TABNAME
```

此查询所标识的表和列可能是其 SQL 语句失败了的视图的基本表。

如果 ALTER TABLE 语句返回了错误，那么在将某列设置为 NOT NULL 时修改包含空值的列的任何行数据，然后再次尝试运行该语句。

联合系统用户：如果原因未知，那么找出使请求失败的数据源中的问题（请参阅《故障诊断指南》以便了解确定失败的数据源时遵循的过程），并检查该数据源的对象定义。切记：不同数据源之间的缺省值（NULL 和 NOT NULL）不必相同。

sqlcode: -407

sqlstate: 23502

18. 执行违反外键约束的操作

----- 输入的命令 -----

```
delete from project where pnumber=301;
```

```
delete from project where pnumber=301
```

DB21034E 该命令被当作 SQL 语句来处理，因为它是无效的“命令行处理器”命令。在 SQL 处理期间，它返回：

SQL0532N 因为关系“ZHAOFENG.WORK.FK_WORK_PNUMBER”

限制删除，所以不能删除父行。 SQLSTATE=23001

SQL0532N 因为关系“ZHAOFENG.WORK.FK_WORK_PNUMBER

”限制删除，所以不能删除父行。

说明：

某个操作尝试删除父表的指定行，但指定行中的父键在引用约束“<约束名>”中有从属行，并且已为关系指定了 NO ACTION 或 RESTRICT 删除规则。

当约束“<约束名>”的删除规则为 NO ACTION 时，如果从属行在语句完成时仍依赖于父键，那么不能删除父表的行。

当约束“<约束名>”的删除规则为 RESTRICT 时，如果父行在删除时有任何从属行，那么不能删除父表的行。

注意，删除可以级联以删除有 NO ACTION 或 RESTRICT 删除规则的从属表中的其他行。因此，约束“<约束名>”可以在与原始删除操作不同的表上。

联合系统用户：约束可以存在于数据源上（如果子表和父表存在于该数据源上）。

某些数据源未向“<约束名>”提供适当的值。在这些情况下，适当的字段包含一些值（如“unknown”），指示实际值未知。

无法执行该语句。表的内容未更改。

用户响应：

检查所有派生表的删除规则以确定并更正问题。涉及的特定表可根据关系“<约束名>”确定。

sqlcode: -532

sqlstate: 23001, 23504

19. 创建一个触发器，当职工参加一个新的项目时，年薪增加2%

----- 输入的命令 -----

```
create trigger update_salary
  after insert
  on work
  referencing new as new_work
  for each row
    update employee
    set esalary = 1.02*esalary
    where enumber = new_work.enumber;
```

```
create trigger update_salary
  after insert
  on work
  referencing new as new_work
  for each row
    update employee
    set esalary = 1.02*esalary
    where enumber = new_work.enumber
```

DB20000I SQL 命令成功完成。

初始employee表如下：

ZHAOFENG_PC - DB2 - COMPANY - ZHAOFENG.EMPLOYEE							
采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”笔记本来更改编辑格式。							
ENAME	ENUMBER	EBIRTHDAY	EADDRESS	ESALARY	MTUMBER	DTUMBER	
张三	101	1981/1/1	北京	10000			201
李四	102	1982/2/2	上海	20000	101		201
王五	103	1983/3/3	南京	30000			202

执行insert into work values(101,301,50);

----- 输入的命令 -----

```
insert into work values(101,301,50);
```

```
insert into work values(101,301,50)
```

DB20000I SQL 命令成功完成。

此时employee如下：

ZHAOFENG_PC - DB2 - COMPANY - ZHAOFENG.EMPLOYEE							
采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”笔记本来更改编辑格式。							
ENAME	ENUMBER	EBIRTHDAY	EADDRESS	ESALARY	MINUMBER	DNUMBER	
张三 ...	101	1981/1/1 ...	北京 ...	10200			201
李四 ...	102	1982/2/2 ...	上海 ...	20000	101		201
王五 ...	103	1983/3/3 ...	南京 ...	30000			202

20. 创建一个用户，将工作表的查询权限赋予该用户

----- 输入的命令 -----

grant select on work to work_select_user;

grant select on work to work_select_user

DB20000I SQL 命令成功完成。

ZHAOFENG_PC - DB2 - COMPANY - WORK_SELECT_USER

数据库	模式	表	索引	视图	表空间	函数	过程	方法	程序包
表	SELECT	INSERT	UPDATE	DELETE	CONTROL	ALTER	INDEX	REFERENCES	
ZHAOFENG.WORK	✓	✗	✗	✗	✗	✗	✗	✗	✗