# 实验三：数据库物理设计

#### 姓名：谢宝玛 学号：1120233506

## 实验目的

创建数据库分区表

体会主键，外键约束

体会索引

权限管理

## 实验过程

### 练习更新、删除主表数据（针对主键属性且子表中可能有参照外键数据）

#### 更新

将学院编号为‘01’的学院改为‘10’，注意教师表和学生表都有学院编号

因为学院编号可能在教师表和学生表里作为外键或关联字段直接使用，直接改可能会导致关联性错误或约束冲突，所以一般采取“新增-修改-删除”三步走策略：

SQL：

-- 第一步：先在学院表新增编号为‘10’的学院，名称和原来的一样

INSERT INTO xyb (ydh, ymc)

SELECT '10', ymc

FROM xyb

WHERE ydh = '01';

-- 第二步：将教师表和学生表中原‘01’编号改为‘10’

UPDATE js

SET ydh = '10'

WHERE ydh = '01';

UPDATE xs

SET ydh = '10'

WHERE ydh = '01';

-- 第三步：删除原来的‘01’学院

DELETE FROM xyb

WHERE ydh = '01';



#### 删除

删除“张三”这名同学

SQL：

-- 先删除该学生在选课表中的记录

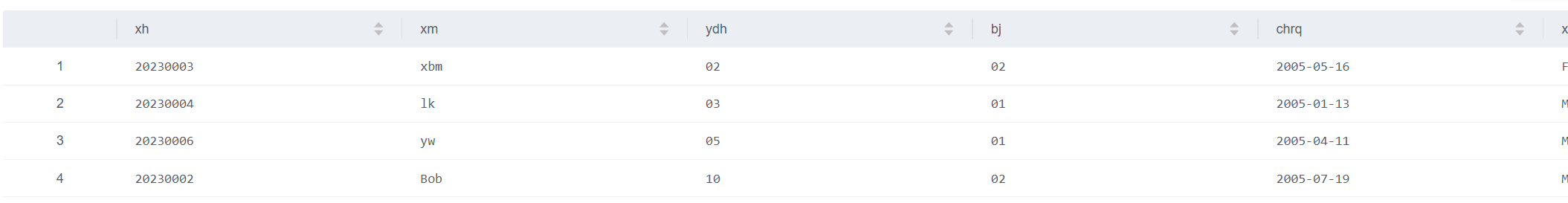
DELETE FROM xk

WHERE xh IN (SELECT xh FROM xs WHERE xm = '张三');

-- 再删除学生表中的记录

DELETE FROM xs

WHERE xm = '张三';



### 练习更新、删除主表数据（针对非主键 属性）

#### 更新

将“车辆学院”改名为“机械学院”

SQL：  
UPDATE xyb

SET ymc = '机械学院'

WHERE ymc = '车辆学院';



### 练习先删除子表数据，再删除主表数据

删除“张三”这名同学

SQL：

-- 先删除该学生在选课表中的记录

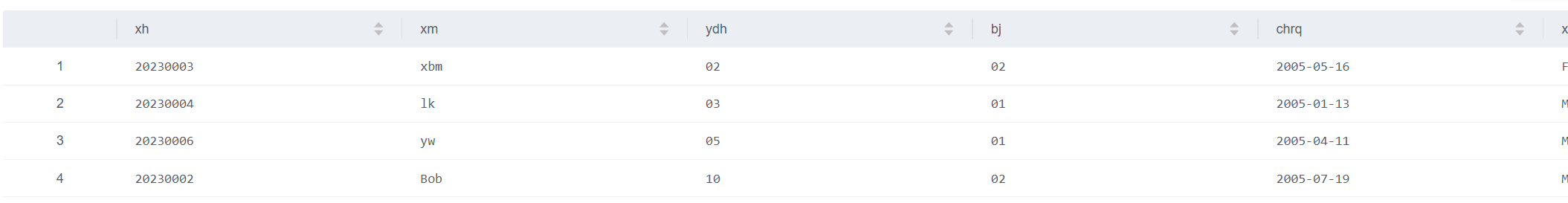
DELETE FROM xk

WHERE xh IN (SELECT xh FROM xs WHERE xm = '张三');

-- 再删除学生表中的记录

DELETE FROM xs

WHERE xm = '张三';



### 使用子查询方式更新、删除数据

将“教授”职称的教师的课程（授课表 sk 中对应 jsbh）在 选课表 xk 里他们授课的所有学生成绩（cj）加 5 分（如果超过 100 则设为 100），要求用**子查询**完成更新。

sk 表中有课程编号（kcbh）和教师编号（bh）js 表中有教师编号（jsbh）和职称（zc）xk 表是选课表，包含学生成绩（cj）、kcbh、jsbh子查询部分需要找出“教授”的教师编号，再根据这个编号去更新 xk 表中的成绩。

SQL：  
UPDATE xk

SET cj = CASE

WHEN cj + 5 > 100 THEN 100

ELSE cj + 5

END

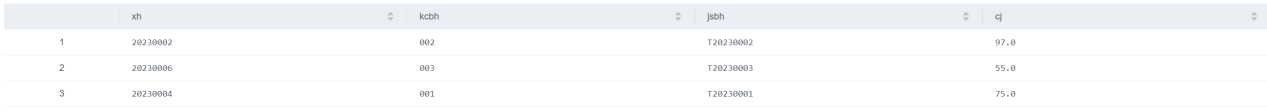
WHERE jsbh IN (

SELECT jsbh

FROM js

WHERE zc = '教授'

);



### 权限管理

#### 以不同身份用户登录数据库建立表，表名一样

##### 1.新增用户xbm

SQL:  
CREATE ROLE "xbm" LOGIN VALID BEGIN '2025-03-23 23:22:11+08:00' VALID UNTIL '2025-07-31 23:22:38+08:00' CREATEDB CREATEROLE INHERIT NOREPLICATION PASSWORD '\*\*\*\*\*\*';

GRANT root TO "xbm" WITH ADMIN OPTION;

/\* Switch to database postgres \*/

GRANT SELECT ON TABLE "pg\_catalog"."pg\_roles" TO "xbm" WITH GRANT OPTION;

我没有采用SQL语句，而是直接在华为的DAS中新增。

##### **2.**建立学院表

先建立一个新的schema “xbm\_student\_info”

再创建表

SQL：  
CREATE TABLE `xyb` (

`ydh` VARCHAR(10) NOT NULL,

`ymc` VARCHAR(50) DEFAULT NULL,

PRIMARY KEY (`ydh`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

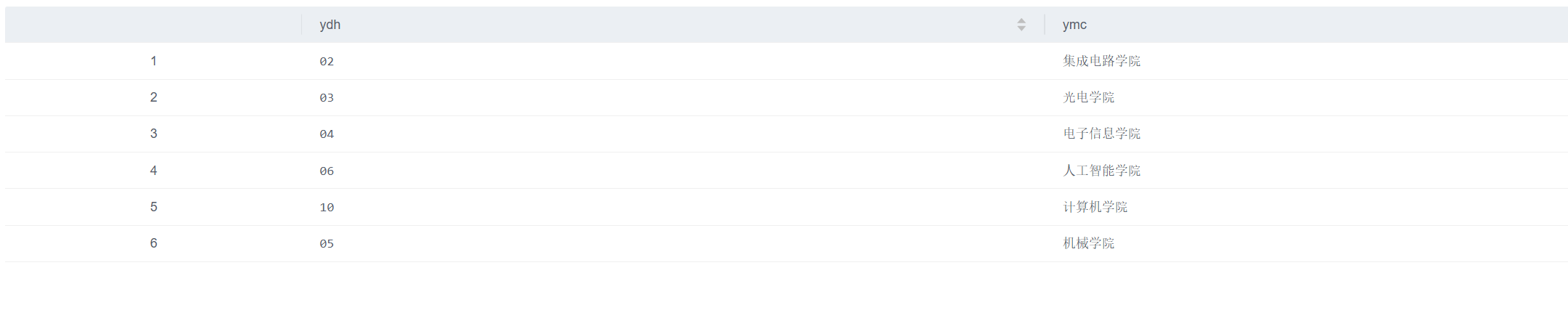
#### 以不同身份用户查询自己与其他用户建立 的表

##### 查询root用户建立的学院表

需要切换到‘student\_info’

SQL：

SELECT \* FROM xyb;



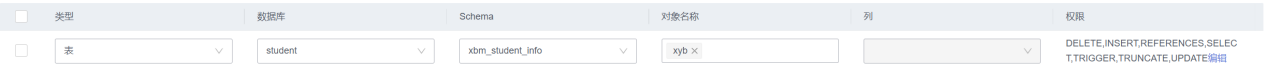
##### **2.**查询xbm用户建立的学院表

需要切换到‘xbm\_student\_info’

SQL:

SELECT \* FROM xyb;

#### 定义授权方案并进行验证

在以上的任务中，xbm用户具有增删改查的权限，并继承了root用户的权限

DELETE,INSERT,REFERENCES,SELECT,TRIGGER,TRUNCATE,UPDATE

修改xbm的权限，只保留select，让xbm用户对xbm\_student\_info只有可读权限。



验证

以xbm用户登录，执行插入操作

SQL:  
INSERT INTO xyb(ydh ,ymc )VALUES

('01','计算机学院');

结果会报错，无权限



但是xbm用户可以执行creat table建表的操作。

## 实验体会

1. 在执行更新，删除操作的时候一定要注意表是否有外键，如果有外键需要对子表先进行更新，删除操作。
2. 共同设计一个数据库的时候需要注意设置非root用户的权限。