数据库实验

实验四

> 姓名：苏柏瑞
> 学号：201600301108
> 班级：泰山学堂
> 实验题目：复制表、修改表结构、修改数据

实验内容

利用oracle管理平台完成对表的结构、数据进行修改，每一个问题可以通过多个SQL语句完成

实验题目

- 将pub用户下表student_41及数据复制到主用户的表test4_01中，使用alter table语句为表增加列：“总成绩:sum_score”。

```sql
CREATE TABLE test4_01 AS SELECT * FROM pub.student_41;
ALTER TABLE test4_01 ADD sum_score int;
UPDATE test4_01 SET sum_score =
(SELECT sum(score) FROM pub.student stu, pub.student_course stuc
WHERE stu.sid = stuc.sid GROUP BY stu.sid HAVING stu.sid = test4_01.sid);
```

- 将pub用户下表student_41及数据复制到主用户的表test4_02中，使用alter table语句为表增加列“平均成绩:avg_score”（小数点后保留1位）。
  利用pub.student_course、pub.course，统计“平均成绩”，四舍五入到小数点后1位

```sql
CREATE TABLE test4_02 AS SELECT * FROM pub.student_41;
ALTER TABLE test4_02 ADD avg_score numeric(4, 1);
UPDATE test4_02 set avg_score =
(SELECT avg(score) FROM pub.student stu, pub.student_course stuc
WHERE stu.sid = stuc.sid GROUP BY stu.sid HAVING stu.sid = test4_02.sid);
```

- 将pub用户下表student_41及数据复制到主用户的表test4_03中，使用alter table语句为表增加列：“总学分:sum_credit”。
  使用update语句,利用pub.student_course、pub.course，统计 “总学分”；
  这是需要注意：成绩及格才能够计算所得学分,一门课多个成绩都及格只计一次学分。

```sql
CREATE TABLE test4_03 AS SELECT * FROM pub.student_41;
ALTER TABLE test4_03 ADD sum_credit int;
UPDATE test4_03 set sum_credit =
(WITH tmp(sid, cid, credit) AS
(SELECT sid, sc.cid, max(credit)
FROM pub.student_course sc, pub.course c WHERE sc.cid = c.cid and score >
= 60 GROUP BY sid, sc.cid)
SELECT sum(credit) from tmp GROUP BY sid HAVING sid = test4_03.sid);
```

- 将pub用户下表student_41及数据复制到主用户的表test4_04中。
  根据列院系名称dname到pub.department找到对应院系编号did，将对应的院系编号回填到院系名称列dname中，如果表中没有对应的院系名称，则列dname中内容不变仍然是原来的内容。

```
CREATE TABLE test4_04 AS (SELECT * FROM pub.student_41);
UPDATE test4_04 set dname =
(SELECT did FROM pub.department dep WHERE dep.dname = test4_04.dname)
WHERE dname in (SELECT dname FROM pub.department);
```

- 将pub用户下表student_41及数据复制到主用户的表test4_05中，使用alter table语句为表增加4个列：“总成绩:sum_score”、 “平均成绩:avg_score”、“总学分:sum_credit”、“院系编号:did varchar(2) ”。
  （1） 利用pub.student_course、pub.course，统计 “总成绩”；
  （2） 利用pub.student_course、pub.course，统计“平均成绩”，四舍五入到小数点后1位；
  （3） 利用pub.student_course、pub.course，统计 “总学分”；
  （4） 根据院系名称到pub.department或者pub.department_41中，找到对应编号，填写到院系编号中，如果都没有对应的院系，则填写为00。

```
CREATE TABLE test4_05 AS SELECT * FROM pub.student_41;
ALTER TABLE test4_05 ADD sum_score int;
ALTER TABLE test4_05 ADD avg_score numeric(4, 1);
ALTER TABLE test4_05 ADD sum_credit int;
ALTER TABLE test4_05 ADD did varchar(2);

UPDATE test4_05 SET sum_score =
(SELECT sum(score) FROM pub.student stu, pub.student_course stuc
WHERE stu.sid = stuc.sid GROUP BY stu.sid HAVING stu.sid = test4_05.sid);

UPDATE test4_05 set avg_score =
(SELECT avg(score) FROM pub.student stu, pub.student_course stuc
WHERE stu.sid = stuc.sid GROUP BY stu.sid HAVING stu.sid = test4_05.sid);

UPDATE test4_05 set sum_credit =
(WITH tmp(sid, cid, credit) AS
(SELECT sid, sc.cid, max(credit)
FROM pub.student_course sc, pub.course c WHERE sc.cid = c.cid and score >
= 60 GROUP BY sid, sc.cid)
SELECT sum(credit) from tmp GROUP BY sid HAVING sid = test4_05.sid);

UPDATE test4_05 set did =
(SELECT did FROM pub.department  WHERE dname = test4_05.dname)
```

```
WHERE dname in (SELECT dname FROM pub.department);

UPDATE test4_05 set did =
(SELECT did FROM pub.department_41  WHERE dname = test4_05.dname)
WHERE dname in (SELECT dname FROM pub.department_41);
```

- 将pub用户下的Student_42及数据复制到主用户的表test4_06中,对表中的数据进行整理，修复那些不规范的数据：
  剔除姓名列中的所有空格；

```
CREATE TABLE test4_06 AS SELECT * FROM pub.student_42;
UPDATE test4_06 SET name = REPLACE(name, ' ', '');
```

- 将pub用户下的Student_42及数据复制到主用户的表test4_07中,对表中的数据进行整理，修复那些不规范的数据：
  对性别列进行规范（需要先确定哪些性别数据不规范，也就是那些和大多数不一样的就是不规范的）；

```
CREATE TABLE test4_07 AS SELECT * FROM pub.student_42;
UPDATE test4_07 SET sex = replace(sex, ' ', '');
update test4_07 SET sex = SUBSTR(sex, 1, 1) WHERE LENGTH(sex) >= 1;
```

- 将pub用户下的Student_42及数据复制到主用户的表test4_08中,对表中的数据进行整理，修复那些不规范的数据：
  对班级列进行规范（需要先确定哪些班级不规范）。

```
CREATE TABLE test4_08 AS SELECT * FROM pub.student_42;
UPDATE test4_08 set class = substr(class, 1, 4) WHERE length(class) > 4;
```

- 将pub用户下的Student_42及数据复制到主用户的表test4_09中,对表中的数据进行整理，修复那些不规范的数据：
  年龄为空值的根据出生日期设置学生年龄（截止到2012年的年龄，即年龄=2012-出生年份），年龄不为空值的不要改变。

```
CREATE TABLE test4_09 AS SELECT * FROM pub.student_42;
UPDATE test4_09 set age = 2012 - extract(year FROM birthday) WHERE age is null;
```

- 将pub用户下的Student_42及数据复制到主用户的表test4_10中,对表中的数据进行整理，修复那些不规范的数据：
  （1） 剔除姓名列中的所有空格；
  （2） 剔除院系名称列中的所有空格；
  （3） 对性别列进行规范（需要先确定哪些性别数据不规范，也就是那些和大多数不一样的就是不规范的）；
  （4） 对班级列进行规范（需要先确定哪些班级不规范）。

（5） 年龄为空值的根据出生日期设置学生年龄（截止到2012年的年龄，即年龄=2012-出生年份），年龄不为空值的不要改变。

```
CREATE TABLE test4_10 AS SELECT * FROM pub.student_42;
UPDATE test4_10 SET name = REPLACE(name, ' ', '');
UPDATE test4_10 SET dname = REPLACE(dname, ' ', '');
UPDATE test4_10 SET sex = replace(sex, ' ', '');
update test4_10 SET sex = SUBSTR(sex, 1, 1) WHERE LENGTH(sex) >= 1;
UPDATE test4_10 set class = substr(class, 1, 4) WHERE length(class) > 4;
UPDATE test4_10 set age = 2012 - extract(year FROM birthday) WHERE age is null;
DROP TABLE test4_05;
update dbtest set test= 4;
```