ДЗ 5

Содержание

1 Условия

Домашнее задание предполагает формулировку выражений реляционной алгебры и соответствующих им SQL запросов над базой данных "деканат".

2 Решение

1. Информация о студентах с заданной оценкой по предмету «Базы данных».

$$\pi_{StudentId,StudentName,GroupName}\left(\sigma_{CourseName='\text{БазыДанных}'\wedge P(Mark)}(((StudentInfo \bowtie Marks) \bowtie GroupInfo) \bowtie CourseInfo)\right)$$

```
select studentid, studentname, groupname
from (((studentinfo natural join marks) natural join groupinfo) natural join courseinfo)
where coursename = 'Базы данных'
and mark > 85;
```

2. Информацию о студентах не имеющих оценки по предмету «Базы данных»:

    1. Среди всех студентов.

$$\pi_{StudentId,StudentName,GroupName}\left(\sigma_{Mark=null\wedge(CourseName=null\vee CourseName='\text{БазыДанных}'}(StudentInfo \rtimes Marks \rtimes CourseInfo \bowtie GroupInfo)\right)\$$$

```
select s.studentid, s.studentname, groupname
from ((studentinfo s left join
       marks m on s.studentid = m.studentid) left join
       courseinfo c on m.courseid = c.courseid) natural join
       groupinfo
where mark is null
and (c.coursename is null or c.coursename = 'Базы данных');
```

    2. Среди студентов, у которых есть этот предмет.

$$\pi_{StudentId,StudentName,GroupName}\left(\sigma_{Mark=null}((\pi_{CourseId}(\sigma_{CourseName='\text{БазыДанных}'}(CourseInfo))) \bowtie WorkingPlan \bowtie StudentInfo \rtimes Marks \bowtie Gro\right.$$

```
select d.studentid, studentname, groupname
from ((((select courseid from courseinfo where coursename = 'Базы данных') c
        natural join workingplan)
       natural join studentinfo) d
      left join marks m using (studentid))
     natural join groupinfo
where mark is null;
```

3. Информация о студентах, имеющих хотя бы одну оценку у заданного лектора.

$$\pi_{StudentId,StudentName,GroupName}((\pi_{LecturerId}(\sigma_{LecturerName='\text{БазыДанных}'}(LecturerInfo))) \bowtie WorkingPlan \bowtie (StudentInfo \rtimes Marks) \bowtie GroupInfo)$$

```
select studentid, studentname, groupname
from (select lecturerid from lecturerinfo where lecturername = 'Георгий Корнеев') k
     natural join
     workingplan
     inner join
     (studentinfo left join marks using (studentid)) using (courseid, groupid)
     natural join
     groupinfo;
```

4. Идентификаторы студентов, не имеющих ни одной оценки у заданного лектора.

$$\pi_{StudentId}(\sigma_{Mark=null}(\sigma_{LecturerName=X}(WorkingPlan \bowtie LecturerInfo) \bowtie Marks \ltimes StudentInfo))$$

```
select studentid
from (select * from (workingplan natural join lecturerinfo) where lecturername = 'Георгий Корнеев') k
     natural join marks
     right join studentinfo using (studentid)
where mark is null;
```

5. Студенты, имеющих оценки по всем предметам заданного лектора.

$$BoolAnd_{HasMark,\{StudentId\}}(\pi_{\{StudentId,HasMark=(Mark==null)\}}(\pi_{\{GroupId,CourseId\}}(WorkingPlan \bowtie LecturerInfo) \bowtie StudentInfo \rtimes StudentInfo))$$

```
select studentid,bool_and(hasMark) as allHasMark
from (select studentid, (case when (mark is null) then false else true end) as hasMark
      from ((select groupid,courseid
```

```
              from (workingplan natural join lecturerinfo)
              where lecturername = 'Георгий Корнеев') k
            natural join studentinfo
            left join marks using (courseid, studentid))) z
group by studentid;
```

6. Для каждого студента имя и предметы, которые он должен посещать.

$$\pi_{StudentName,CourseName}(StudentInfo \bowtie GroupInfo \bowtie WorkingPlan \bowtie CourseInfo)$$

```
select distinct studentname, coursename
from studentinfo natural join groupinfo natural join workingplan natural join courseinfo
order by studentname;
```

http://stackoverflow.com/a/43944 В Postgre 9 можно аггрегировать по строчкам через запятую (`string_agg/array_agg`), если это то, что требовалось. В дальнейшем я буду предполагать, что нет, потому что функция достаточно специфичная.

7. По лектору все студенты, у которых он хоть что-нибудь преподавал.

$$\pi_{LecturerName,StudentName}(LecturerInfo \bowtie WorkingPlan \bowtie GroupInfo \bowtie CourseInfo)$$

```
select distinct lecturername, studentname
from (lecturerinfo natural join workingplan natural join groupinfo natural join studentinfo);
```

8. TODO
9. TODO
10. Средний балл студента. Будем считать, что средний балл – средний по всем предметам, который студент изучеает. Оценки нет => оценка 0. Примечание: интерфейс функции в терминах алгебры подразумевает выкидывание всех атрибутов кроме тех, по которым итерируемся. Будем считать, что они не выкидываются, иначе запрос придется копировать дважды.

    1. По идентификатору.

$$Avg_{Average,\{markReal\}}(\varepsilon_{markReal=ifnullmarkthen0elsemark}(\pi_{StudentId,StudentName}(\sigma_{studentid=X}(StudentInfo \bowtie GroupInfo \bowtie WorkingPlan \rtimes Marks))))$$

```
select avg(markReal) as average
from (select studentid,
             studentname,
             (case when (mark is null) then 0 else mark end) as markReal
      from (studentinfo natural join
            groupinfo natural join
            workingplan left join
            marks using (studentid,courseid))
      where studentid = 1) z
group by studentid, studentname;
```

    2. Для каждого студента.

$$Avg_{Average,\{markReal\}}(\varepsilon_{markReal=ifnullmarkthen0elsemark}(\pi_{StudentId,StudentName}(StudentInfo \bowtie GroupInfo \bowtie WorkingPlan \rtimes Marks)))$$

```
select studentid, studentname, avg(markReal) as average
from (select studentid,
             studentname,
             (case when (mark is null) then 0 else mark end) as markReal
      from (studentinfo natural join
            groupinfo natural join
            workingplan left join
            marks using (studentid,courseid))) z
group by studentid, studentname;
```

11. Средний балл средних баллов студентов каждой группы.

$$Avg_{AverageGroup,\{AverageStudent\}}($$
$$\pi_{GroupId,AverageStudent}(Avg_{AverageStudent,\{markReal\}}(\varepsilon_{markReal=ifnullmarkthen0elsemark}(\pi_{StudentId,GroupId}(StudentInfo \bowtie GroupInfo \bowtie WorkingPlan \rtimes Ma$$

```
select groupid, avg(averageStudent) as averageGroup
from (select groupid, avg(markReal) as averageStudent
      from (select studentid,
                   groupid,
                   (case when (mark is null) then 0 else mark end) as markReal
            from (studentinfo natural join
                  groupinfo natural join
                  workingplan left join
                  marks using (studentid,courseid))) z
      group by studentid, groupid) y
group by groupid;
```

12. TODO

Автор: Михаил Волхов M3438
Created: 2016-11-01 Tue 14:57
Validate