Волков Александр, М3437

Домашнее задание 5

Структура базы данных «Университет»:

- *Students(StudentId, StudentName, GroupId)*
- *Groups(GroupId, GroupName)*
- *Courses(CourseId, CourseName)*
- *Lecturers(LecturerId, LecturerName)*
- *Plan(GroupId, CourseId, LecturerId)*
- *Marks(StudentId, CourseId, Mark)*

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

$$\sigma_{Mark=mark}(\sigma_{CourseName=\text{Базы данных}}(Courses) \rtimes Marks) \rtimes Students$$

```
select StudentId, StudentName, GroupId
from (select StudentId, Mark
        from
                (select CourseId
                from Courses
                where CourseName = 'Базы данных') as C
        natural join Marks
where Mark = 60.0
) as R natural join Students
```

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

а) среди всех студентов

$$Students - ((\sigma_{CourseName=\text{Базы данных}}(Courses) \rtimes Marks) \rtimes Students)$$

```
select StudentId, StudentName, GroupId from Students except all
        select StudentId, StudentName, GroupId from
                (select StudentId from
                        (select CourseId from Courses
                        where CourseName = 'Базы данных') as C
                natural join Marks
        ) as R natural join Students
```

б) среди студентов, у которых есть этот предмет

$$(Students \ltimes (Plan \ltimes (\sigma_{CourseName=\text{Базы данных}}(Courses)))) - ((\sigma_{CourseName=\text{Базы данных}}(Courses) \rtimes Marks) \rtimes Students)$$

```
select StudentId, StudentName, GroupId from Students
        natural join Plan natural join Courses
        where CourseName = 'Базы данных' except all
                select StudentId, StudentName, GroupId from
                        (select StudentId from
                                (select CourseId from Courses
                                where CourseName = 'Базы данных') as C
                        natural join Marks
                ) as R natural join Students
```

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

$$Students \ltimes ((\pi_{LecturerId}(\sigma_{LecturerName=lecturer}(Lecturers)) \bowtie Plan) \bowtie Marks)$$

```
select StudentId, StudentName, GroupId from Students
        natural join
        (select LecturerId from Lecturers where LecturerName = 'Георгий Корнеев' ) as C
        natural join Plan
        natural join Marks
```

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

$$\pi_{StudentId}(Students) - \pi_{StudentId}((\pi_{LecturerId}(\sigma_{LecturerName=lecturer}(Lecturers)) \bowtie Plan) \bowtie Marks)$$

```
select StudentId from Students except all select StudentId from Students
        natural join
        (select LecturerId from Lecturers where LecturerName = 'Корнеев Георгий' ) as C
        natural join Plan
        natural join Marks
```

5. Студентов, имеющих оценки по всем предметам заданного лектора

$$Students \ltimes (\pi_{StudentId, CourseId}(Marks) \div (\pi_{CourseId}((\pi_{LecturerId}(\sigma_{LecturerName=lecturer}(Lecturers)) \bowtie Plan))))$$

```
select StudentId, StudentName, GroupId from Students natural join (
        select distinct StudentId from Marks
        where not exists
            (select CourseId from ((select LecturerId from Lecturers where LecturerName = 'Корнеев Георгий') as L natural join Plan) as C where not exists
            (select StudentId from Marks as M where M.StudentId = Marks.StudentId and M.CourseId = C.CourseId))
        ) as R
```

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

$$\pi_{StudentName, CourseName}(Students \bowtie Plan \bowtie Courses)$$

```
select StudentName, CourseName from
    Students natural join Plan natural join Courses
```

7. По лектору всех студентов, у которых он хоть что-нибудь преподавал

$$(\pi_{LecturerId}(\sigma_{LecturerName=lecturer}(Lecturers)) \bowtie Plan) \bowtie Students$$

```
select StudentId, StudentName, GroupId from Students natural join
                ((select LecturerId from Lecturers where LecturerName = 'Корнеев Георгий' ) as C natural join Plan) as L
```

8.Пары студентов, такие, что все сданные первым студентом предметы сдал и второй студент.

$$(\pi_{StudentName, CourseId}(\sigma_{StudentName=X \wedge Mark \geq 60}(Students \bowtie Marks))) \divideontimes (\pi_{StudentName, CourseId}(\sigma_{StudentName=Y \wedge Mark \geq 60}(Students \bowtie Marks)))$$

9.Такие группы и предметы, что все студенты группы сдали предмет.

$$(\pi_{CourseId, StudentId}(\sigma_{Mark \geq 60} Marks) \divideontimes \pi_{StudentId, Groupid}(Students)) \bowtie Groups \bowtie Courses$$

10.Средний балл студента

а) по идентификатору

$$avg_{Mark,\varnothing}(\sigma_{StudentId=?}(Marks))$$

```
select avg(Mark) from Marks where StudentId=?
```

б) для каждого студента

$$avg_{Mark,\{StudentId\}}(Marks) \bowtie (\pi_{StudentId,StudentName}(Students))$$

```
select avg(Mark), StudentId, StudentName from
    Marks natural join Students
  group by
    StudentId, StudentName
```

11. Средний балл средних баллов студентов каждой группы.

$$avg_{Mark,\{GroupId\}}(avg_{Mark,\{StudentId\}}(Marks) \bowtie \pi_{StudentId,GroupId}(Students)) \bowtie Groups$$

```
select avg(Mark), GroupId, GroupName from
    (select avg(Mark) as Mark, StudentId, GroupId from Marks natural join Students group by StudentId, GroupId) as R
    natural join Groups
  group by
    GroupId, GroupName
```

12. Для каждого студента число предметов, которые у него были, число сданных предметов и число несданных предметов.

$$\varepsilon_{Failed=Total-Passed}(count_{Total,\{StudentId\}}(Students \bowtie Plan) \bowtie count_{Passed,\{StudentId\}}(\sigma_{Mark \geq 60}(Marks) \ltimes = Students)) \bowtie Students$$

```
select StudentId, StudentName, Total, Passed, Total - Passed as Failed from
    (select count(CourseId) as Total, StudentId from Students natural join Plan
      group by
        StudentId
    ) as R1
    natural join
    (select count(CourseId) as Passed, StudentId from (select * from Marks where Mark >= 60) as R3
        right join Students using (StudentId)
     group by
        StudentId
    ) as R2 natural join Students
```