

Королева Алиса, М3437

Структура базы данных «Деканат»:

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

В таблицах я считаю, что вносится информация о положительной оценке у студента (≥ 3), оценки не в виде баллов, а в виде оценки в пятибалльной шкале, то есть это что-то вроде студенческой зачетки

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

Запрос в терминах реляционных исчислений:

```
select StudentId, StudentName, GroupId
from Students
where  $\exists$ Course  $\exists$ Mark (Course.CourseName = 'Базы данных'  $\wedge$  Mark.Mark='5'  $\wedge$ 
Mark.StudentId = Students.StudentId  $\wedge$  Mark.CourseId = Course.CourseId)
```

Запрос в терминах Datalog:

```
R(StudentId, StudentName, GroupId) :- Students(StudentId, StudentName, GroupId),
Courses(CourseId, CourseName), Marks(StudentId, CourseId, Mark), CourseName = 'Базы
данных', Mark='5'
```

SQL-запрос:

```
select StudentID, StudentName, GroupID from Students
where StudentID in
(select Marks.StudentID from Marks where
Marks.MarkValue = 5 and
Marks.CourseID in
(select Courses.CourseID from Courses where
Courses.CourseName = 'Базы данных'));
```

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

а). среди всех студентов

Запрос в терминах реляционных исчислений:

```
select StudentId, StudentName, GroupId
from Students
where  $\neg \exists$ Mark  $\exists$ Course (Course.CourseName = 'Базы данных'  $\wedge$ 
Mark.StudentId = Students.StudentId  $\wedge$  Mark.CourseId = Course.CourseId))
```

```
select StudentId, StudentName, GroupId
```

from Students
where \exists Course \forall Mark (Course.CourseName = 'Базы данных' \wedge
Mark.StudentId \neq Students.StudentId \wedge Mark.CourseId = Course.CourseId))

Королева Алиса, М3437

Запрос в терминах Datalog:

R(StudentId, StudentName, GroupId) :- Students(StudentId, StudentName, GroupId),
Courses(CourseId, CourseName), \neg Marks(StudentId, CourseId, _), CourseName = 'Базы
данных'

SQL-запрос:

```
select StudentID, StudentName, GroupID from Students
where StudentID not in
(select Marks.StudentID from Marks where
Marks.CourseID in
(select Courses.CourseID from Courses where
Courses.CourseName = 'Базы данных'));
```

б). среди всех студентов, у которых есть этот предмет

Запрос в терминах реляционных исчислений:

```
select StudentId, StudentName, GroupId
from Students
where  $\neg \exists$ Mark  $\exists$ Course  $\exists$ Plan (Course.CourseName = 'Базы данных'  $\wedge$   
Mark.StudentId = Students.StudentId  $\wedge$  Mark.CourseId = Course.CourseId  $\wedge$  Plan.CourseId  
= Course.CourseId  $\wedge$  Plan.GroupId = Students.GroupId))
```

```
select StudentId, StudentName, GroupId
from Students
where  $\exists$ Course  $\exists$ Plan  $\forall$ Mark (Course.CourseName = 'Базы данных'  $\wedge$   
Mark.CourseId = Course.CourseId  $\wedge$  Plan.CourseId = Course.CourseId  $\wedge$  Plan.GroupId =  
Students.GroupId  $\wedge$  Mark.StudentId  $\neq$  Students.StudentId))
```

Запрос в терминах Datalog:

R(StudentId, StudentName, GroupId) :- Students(StudentId, StudentName, GroupId),
Courses(CourseId, CourseName), \neg Marks(StudentId, CourseId, _), Plan(GroupId, CourseId,
_), CourseName = 'Базы данных'

Королева Алиса, М3437

SQL-запрос:

```
select StudentID, StudentName, GroupID from Students where
    Students.StudentID not in
    (select Marks.StudentID from Marks where
        Marks.StudentID = Students.StudentID and
        Marks.CourseID in
        (select Courses.CourseID from Courses where
            Courses.CourseName = 'Базы данных'))
and Students.GroupID in
    (select GroupID from Plan where
        Plan.CourseID in
        (select Courses.CourseID from Courses where
            Courses.CourseName = 'Базы данных')));
```

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

Запрос в терминах реляционных исчислений:

```
select StudentId, StudentName, GroupId
from Students
where  $\exists$ Lecturers  $\exists$ Mark  $\exists$ Plan(Lecturers.LecturerName = 'Георгий Корнеев'  $\wedge$ 
    Plan.LecturerId = Lecturers.LecturerId  $\wedge$  Plan.CourseId = Mark.CourseId  $\wedge$ 
    Mark.StudentId = Students.StudentId)
```

Запрос в терминах Datalog:

```
R(StudentId, StudentName, GroupId) :- Students(StudentId, StudentName, GroupId),
    Marks(StudentId, CourseId, LecturerId), Plan(GroupId, CourseId, LecturerId),
    Lecturers(LecturerId, LecturerName), LecturerName = 'Георгий Корнеев'
```

SQL-запрос:

```
select StudentID, StudentName, GroupID from Students
where StudentID in
    (select Marks.StudentID from Marks where
        Marks.CourseID in
        (select CourseID from Plan where
            LecturerID in
            (select LecturerID from Lecturers where
                Lecturers.LecturerName = 'Георгий Корнеев'))));
```

Королева Алиса, М3437

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

Запрос в терминах реляционных исчислений:

```
select StudentId
from Students
where ¬∃Mark ∃Lecturers ∃Plan (Lecturers.LecturerName = 'Георгий Корнеев' ∧
    Plan.LecturerId = Lecturers.LecturerId ∧ Plan.CourseId = Mark.CourseId ∧
    Mark.StudentId = Students.StudentId)
```

Запрос в терминах Datalog:

```
R(StudentId) :- Students(StudentId, _, _), ¬Marks(StudentId, CourseId, LecturerId),
    Plan(_, CourseId, LecturerId), Lecturers(LecturerId, LecturerName),
    LecturerName = 'Георгий Корнеев'
```

SQL-запрос:

```
select StudentID from Students
where Students.StudentID not in
(select Marks.StudentID from Marks where
    Marks.CourseID in
    (select CourseID from Plan where
        LecturerID in
        (select LecturerID from Lecturers where
            Lecturers.LecturerName = 'Георгий Корнеев'))));
```

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

Запрос в терминах реляционных исчислений:

```
select StudentId
from Students
where ∃Lecturers ∀Plan ∃Mark (Lecturers.LecturerName = 'Георгий Корнеев' ∧
    Plan.LecturerId = Lecturers.LecturerId ∧ Plan.CourseId = Mark.CourseId ∧
    Mark.StudentId = Students.StudentId)
```

Запрос в терминах Datalog:

```
AllCoursesOfLecturer(CourseId) :- Plan(_, CourseId, LecturerId), Lecturers(LecturerId,
    LecturerName), LecturerName = 'Георгий Корнеев'
StudentsWithoutMark(StudentId) :- Students(StudentId, _, _), ¬Marks(StudentId, CourseId,
    LecturerId), AllCoursesOfLecturer(CourseId)
```

$R(\text{StudentId}) :- \text{Students}(\text{StudentId}, _, _), \neg \text{StudentsWithoutMark}(\text{StudentId})$

Королева Алиса, М3437

SQL-запрос:

```
select StudentID from Students where
  not exists (select * from (select distinct CourseID from Plan where
    LecturerID in
      (select LecturerID from Lecturers where
        Lecturers.LecturerName = 'Георгий Корнеев') and
    CourseID not in (
      select distinct Marks.CourseID from Marks where
        Marks.StudentID = Students.StudentID and
        Marks.CourseID in (select distinct CourseID from Plan where
          LecturerID in
            (select LecturerID from Lecturers where
              Lecturers.LecturerName = 'Георгий Корнеев')))) as aaa);
```

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

Запрос в терминах реляционных исчислений:

```
select StudentName, CourseName
from Students, Courses
where  $\exists \text{Plan} (\text{Plan.GroupId} = \text{Students.GroupId} \wedge \text{Plan.CourseId} = \text{Courses.CourseId})$ 
```

Запрос в терминах Datalog:

$R(\text{StudentName}, \text{CourseName}) :- \text{Students}(_, \text{StudentName}, \text{GroupId}), \text{Plan}(\text{GroupId}, \text{CourseId}, _),$
 $\text{Courses}(\text{CourseId}, \text{CourseName})$

SQL-запрос:

```
select Students.StudentName, Courses.CourseName from Students, Courses, Plan
where
  Courses.CourseID = Plan.CourseID and
  Students.GroupID = Plan.GroupID;
```

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

Запрос в терминах реляционных исчислений:

```
select LecturerName, StudentName
from Lecturers, Students
where  $\exists \text{Plan} (\text{Plan.LecturerId} = \text{Lecturers.LecturerId} \wedge \text{Plan.GroupId} = \text{Students.GroupId})$ 
```

Запрос в терминах Datalog:

$R(\text{LecturerName}, \text{StudentName}) :- \text{Students}(_, \text{StudentName}, \text{GroupId}),$

Plan(GroupID, _, LecturerID), Lecturers(LecturerID, LecturerName)

Королева Алиса, М3437

SQL-запрос:

```
select distinct Lecturers.LecturerName, Students.StudentName from Students, Lecturers, Plan
where
    Lecturers.LecturerID = Plan.LecturerID and
    Students.GroupID = Plan.GroupID;
```

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

Запрос в терминах реляционных исчислений:

```
select S1.StudentName, S2.StudentName
from Students as S1, Students as S2
where  $\forall$ (Mark as M1)  $\exists$ (Mark as M2) (S1.StudentId = M1.StudentId  $\wedge$  M1.CourseId =
    M2.CourseId  $\wedge$  S2.StudentId = M2.StudentId)
```

Запрос в терминах Datalog:

```
AllPassedByStudent(StudentId, CourseId) :- Students(StudentId, _, _),
    Marks(StudentId, CourseId, _)
R(SN1, SN2) :- Students(SId1, SN1, _), AllPassedByStudent(SId1, CourseId),
    Students(SId2, SN2, _), Marks(SId2, CourseId, _)
```

SQL-запрос:

```
select S1.StudentName, S2.StudentName from Students as S1, Students as S2
where not exists
    (select * from Marks where
        Marks.StudentID = S1.StudentID and
        Marks.CourseID not in (select CourseID from Marks where Marks.StudentID =
            S2.StudentID))
and exists (select CourseID from Marks where Marks.StudentID = S1.StudentID);
```