

Домашнее задание № 5

Кокорин Илья, М3439

20 октября 2019 г.

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

Считаем заданной оценкой диапазон оценок.

$\pi_{StudentId, StudentName, GroupName}(\sigma_{CourseName='Базы данных' \wedge Mark \geq ? \wedge Mark \leq ?}(Students \bowtie Marks \bowtie Courses \bowtie Groups))$

```
SELECT DISTINCT Students.StudentId, Students.StudentName, Groups.GroupName
FROM Students
      NATURAL JOIN Marks
      NATURAL JOIN Courses
      NATURAL JOIN Groups
WHERE Courses.CourseName = 'Базы данных'
AND Marks.Mark <= ? AND Marks.Mark >= ?;
```

2 Информацию о студентах не имеющих оценки по предмету «Базы данных» среди всех студентов

$\pi_{StudentId, StudentName, GroupName}((\pi_{StudentId, StudentName, GroupId}(\pi_{StudentId, StudentName, GroupId}(Students) - \pi_{StudentId, StudentName, GroupId}(\sigma_{CourseName='Базы данных'}(Students \bowtie Marks \bowtie Courses)))) \bowtie Groups)$

```
WITH CorrectCourses AS (
  SELECT Courses.CourseId
  FROM Courses
  WHERE Courses.CourseName = 'Базы данных'
)
SELECT DISTINCT CorrectStudents.StudentId,
                 CorrectStudents.StudentName,
                 Groups.GroupName
FROM (SELECT DISTINCT Students.StudentId,
                     Students.StudentName,
                     Students.GroupId
      FROM Students EXCEPT ALL (SELECT DISTINCT Students.StudentId,
                                                Students.StudentName,
                                                Students.GroupId
                                FROM Students
                                NATURAL JOIN Marks
                                NATURAL JOIN CorrectCourses)) AS CorrectStudents
NATURAL JOIN Groups;
```

3 Информацию о студентах не имеющих оценки по предмету «Базы данных» среди студентов, у которых есть этот предмет

$$\pi_{StudentId, StudentName, GroupName}((\pi_{StudentId, StudentName, GroupId}(\sigma_{CourseName='Базы данных'}(Students \bowtie Plan \bowtie Courses)) - \pi_{StudentId, StudentName, GroupId}(\sigma_{CourseName='Базы данных'}(Students \bowtie Marks \bowtie Courses))) \bowtie Groups)$$

```
WITH CorrectCourses AS (
    SELECT Courses.CourseId
    FROM Courses
    WHERE Courses.CourseName = 'Базы данных'
)
SELECT DISTINCT CorrectStudents.StudentId,
                CorrectStudents.StudentName,
                Groups.GroupName
FROM (SELECT DISTINCT Students.StudentId,
                    Students.StudentName,
                    Students.GroupId
    FROM Students
        NATURAL JOIN Plan
        NATURAL JOIN CorrectCourses
    EXCEPT ALL (SELECT DISTINCT Students.StudentId,
                                Students.StudentName,
                                Students.GroupId
    FROM Students
        NATURAL JOIN Marks
        NATURAL JOIN CorrectCourses)) AS CorrectStudents
    NATURAL JOIN Groups;
```

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

$$\pi_{StudentId, StudentName, GroupName}(\sigma_{LecturerId=?}(Students \bowtie Groups \bowtie Marks \bowtie Plan))$$

```
SELECT DISTINCT Students.StudentId,
                Students.StudentName,
                Groups.GroupName
FROM Students
    NATURAL JOIN Groups
    NATURAL JOIN Marks
    NATURAL JOIN Plan
WHERE Plan.LecturerId = ?;
```

5 Идентификаторы студентов, не имеющих ни одной оценки у заданного лектора

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

```
SELECT DISTINCT Students.StudentId
FROM Students EXCEPT ALL (
    SELECT DISTINCT Students.StudentId
```

```

FROM Students
    NATURAL JOIN Marks
    NATURAL JOIN Plan
WHERE Plan.LecturerId = ?
);

```

6 Студентов, имеющих оценки по всем предметам заданного лектора

Будем требовать, чтобы у студента были оценки по всем предметам, которые заданный лектор когда-либо вёл в какой-либо группе. Также считаем, что если заданный лектор вообще не ведёт предметов, то оценки по всем его предметам есть у всех студентов.

$$\pi_{StudentId, StudentName, GroupName, CourseId, LecturerId}(Students \bowtie Groups \bowtie Marks \bowtie Plan) \div \pi_{LecturerId, CourseId}(\sigma_{LecturerId=?}(Plan))$$

```

WITH StudentsWithCourses AS (
    SELECT DISTINCT Students.StudentId,
                     Students.StudentName,
                     Groups.GroupName,
                     Marks.CourseId,
                     Plan.LecturerId
    FROM Students
        NATURAL JOIN Groups
        LEFT OUTER JOIN Marks USING (StudentId)
        LEFT OUTER JOIN Plan USING (GroupId, CourseId)
),
ProperLecturerCourses AS (
    SELECT DISTINCT Plan.CourseId,
                     Plan.LecturerId
    FROM Plan
    WHERE LecturerId = ?
)
SELECT DISTINCT StudentsWithCourses.StudentId,
                 StudentsWithCourses.StudentName,
                 StudentsWithCourses.GroupName
FROM StudentsWithCourses EXCEPT ALL (
    SELECT DISTINCT StudentsWithNotAllCourses.StudentId,
                     StudentsWithNotAllCourses.StudentName,
                     StudentsWithNotAllCourses.GroupName
    FROM (
        (SELECT DISTINCT StudentsWithCourses.StudentId,
                         StudentsWithCourses.StudentName,
                         StudentsWithCourses.GroupName,
                         ProperLecturerCourses.CourseId,
                         ProperLecturerCourses.LecturerId
         FROM StudentsWithCourses
          CROSS JOIN ProperLecturerCourses)
    EXCEPT
    ALL
    (SELECT DISTINCT StudentsWithCourses.StudentId,
                     StudentsWithCourses.StudentName,
                     StudentsWithCourses.GroupName,
                     StudentsWithCourses.CourseId,
                     StudentsWithCourses.LecturerId

```

```

        FROM StudentsWithCourses)
    ) AS StudentsWithNotAllCourses
);

```

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

Студент должен посещать все предметы, которые есть в учебном плане его группы, за исключением тех, по которым он уже получил оценку.

$$\begin{aligned}
 & \pi_{StudentId, StudentName, CourseName} (\\
 & \quad \pi_{StudentId, StudentName, CourseId, CourseName} (Students \bowtie Plan \bowtie Courses) \quad - \\
 & \quad \pi_{StudentId, StudentName, CourseId, CourseName} (Students \bowtie Marks \bowtie Courses) \\
 &)
 \end{aligned}$$

```

SELECT DISTINCT NotMarkedCourses.StudentId,
                NotMarkedCourses.StudentName,
                NotMarkedCourses.CourseName
FROM (
    (SELECT DISTINCT Students.StudentId,
                     Students.StudentName,
                     Courses.CourseId,
                     Courses.CourseName
     FROM Students
      NATURAL JOIN Plan
      NATURAL JOIN Courses)
  EXCEPT
  ALL
  (
    SELECT DISTINCT Students.StudentId,
                     Students.StudentName,
                     Courses.CourseId,
                     Courses.CourseName
    FROM Students
     NATURAL JOIN Marks
     NATURAL JOIN Courses
  )
) AS NotMarkedCourses;

```

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

Считаем, что лектор преподавал у студента, если студент учится в группе, в которой преподаватель ведёт хотя бы один предмет.

8.1 Для каждого лектора

$$\pi_{LecturerId, LecturerName, StudentId, StudentName, GroupName} (Plan \bowtie Lecturers \bowtie Students \bowtie Groups)$$

```

SELECT DISTINCT Lecturers.LecturerId,
                Lecturers.LecturerName,
                Students.StudentId,

```

```

        Students.StudentName,
        Groups.GroupName
FROM Plan
    NATURAL JOIN Lecturers
    NATURAL JOIN Students
    NATURAL JOIN Groups;

```

8.2 Для заданного лектора

$\pi_{LecturerId, LecturerName, StudentId, StudentName, GroupName}(\sigma_{LecturerId=?}(Plan \bowtie Students \bowtie Groups) \bowtie Lecturers)$

```

SELECT DISTINCT Lecturers.LecturerId,
               Lecturers.LecturerName,
               LecturerStudents.StudentId,
               LecturerStudents.StudentName,
               LecturerStudents.GroupName
FROM (SELECT Students.StudentId,
            Students.StudentName,
            Groups.GroupName,
            Plan.LecturerId
     FROM Plan
        NATURAL JOIN Students
        NATURAL JOIN Groups
     WHERE Plan.LecturerId = ?) AS LecturerStudents
     NATURAL JOIN Lecturers;

```

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

Заметим, что если первый студент не сдал ни одного предмета, то в пару ему можно поставить любого студента. (назовём такие пары тривиальными, остальные - нетривиальными)

Также выкинем такие пары, которые составлены из одного и того же студента.

$$StudentsWithPassedCourses = \pi_{StudentId, CourseId}(\sigma_{Mark \geq 60}(Students \bowtie Marks))$$

$$NonTrivialPairs = \rho_{StudentId=StudentId2}(StudentsWithPassedCourses) \times \rho_{StudentId=StudentId1}(StudentsWithPassedCourses)$$

$$TrivialPairs = \rho_{StudentId=StudentId1}(\pi_{StudentId}(Students) - \pi_{StudentId}(Students \bowtie Marks)) \times \rho_{StudentId=StudentId2}(\pi_{StudentId}(Students))$$

$$\rho_{StudentName=StudentName1}((\sigma_{StudentId1 \neq StudentId2}(NonTrivialPairs \cup TrivialPairs)) \bowtie_{StudentId1=StudentId} Students) \bowtie_{StudentId2=StudentId} Students)$$

```

WITH StudentsWithPassedCourses AS (
    SELECT DISTINCT Students.StudentId, Marks.CourseId
    FROM Students
        NATURAL JOIN Marks
    WHERE Marks.Mark >= 60
),
StudentsWithPassedCourses1 AS (
    SELECT DISTINCT StudentsWithPassedCourses.StudentId AS StudentId1,

```

```

        StudentsWithPassedCourses.CourseId
    FROM StudentsWithPassedCourses
),
StudentsWithPassedCourses2 AS (
    SELECT DISTINCT StudentsWithPassedCourses.StudentId AS StudentId2,
        StudentsWithPassedCourses.CourseId
    FROM StudentsWithPassedCourses
),
Students1 AS (
    SELECT DISTINCT StudentsWithPassedCourses1.StudentId1 FROM StudentsWithPassedCourses1
),
Students2 AS (
    SELECT DISTINCT StudentsWithPassedCourses2.StudentId2 FROM StudentsWithPassedCourses2
),
AllStudentsPairs AS (
    SELECT DISTINCT Students1.StudentId1,
        Students2.StudentId2
    FROM Students1
        CROSS JOIN Students2
),
IncorrectPairs AS (
    SELECT DISTINCT Pairs.StudentId1, Pairs.StudentId2
    FROM ((SELECT DISTINCT Students2.StudentId2,
        StudentsWithPassedCourses1.StudentId1,
        StudentsWithPassedCourses1.CourseId
        FROM Students2
            CROSS JOIN StudentsWithPassedCourses1)
    EXCEPT
    ALL
    (SELECT DISTINCT StudentsWithPassedCourses2.StudentId2,
        StudentsWithPassedCourses1.StudentId1,
        StudentsWithPassedCourses1.CourseId
    FROM StudentsWithPassedCourses2
        NATURAL JOIN StudentsWithPassedCourses1)
    ) AS Pairs
),
NonTrivialPairs AS (
    SELECT DISTINCT Pairs.StudentId1, Pairs.StudentId2
    FROM (SELECT DISTINCT AllStudentsPairs.StudentId1,
        AllStudentsPairs.StudentId2
        FROM AllStudentsPairs EXCEPT ALL
        SELECT DISTINCT IncorrectPairs.StudentId1,
            IncorrectPairs.StudentId2
        FROM IncorrectPairs) AS Pairs),
AllStudents AS (
    SELECT DISTINCT Students.StudentId
    FROM Students
),
StudentsWithoutPassedCourses AS (
    SELECT DISTINCT Studs.StudentId
    FROM (
        (SELECT DISTINCT AllStudents.StudentId
        FROM AllStudents)
    EXCEPT
    ALL
    (

```

```

        SELECT DISTINCT StudentsWithPassedCourses.StudentId
        FROM StudentsWithPassedCourses
    )
    ) AS Studs
),
TrivialPairs AS (
    SELECT DISTINCT StudentsWithPassedCourses.StudentId AS StudentId1,
                   Students.StudentId AS StudentId2
    FROM StudentsWithPassedCourses
    CROSS JOIN Students
),
AllPairs AS (
    SELECT DISTINCT Pairs.StudentId1,
                   Pairs.StudentId2
    FROM (
        (SELECT StudentId1, StudentId2 FROM NonTrivialPairs)
        UNION
        DISTINCT
        (
            SELECT StudentId1, StudentId2
            FROM TrivialPairs
        )
    )
    ) AS Pairs
    WHERE Pairs.StudentId1 <> Pairs.StudentId2
),
WithNames1 AS (
    SELECT DISTINCT AllPairs.StudentId1,
                   AllPairs.StudentId2,
                   Students.StudentName AS StudentName1
    FROM AllPairs
    INNER JOIN Students ON Students.StudentId = AllPairs.StudentId1
)
SELECT DISTINCT WithNames1.StudentId1,
               WithNames1.StudentName1,
               WithNames1.StudentId2,
               Students.StudentName AS StudentName2
FROM WithNames1
    INNER JOIN Students ON Students.StudentId = WithNames1.StudentId2;

```

10 Такие группы и предметы, что все студенты группы сдали предмет

$$\begin{aligned}
 & \pi_{Groupid, GroupName, CourseId, CourseName}((\\
 & \pi_{CourseId, StudentId}(\sigma_{Mark \geq 60}(Students \bowtie Marks)) \quad * \\
 & \pi_{StudentId, GroupId}(Students) \\
 &) \bowtie Groups \bowtie Courses)
 \end{aligned}$$

```

WITH StudentsWithPassedCourses AS (
    SELECT DISTINCT Marks.CourseId,
                   Students.StudentId
    FROM Students
    NATURAL JOIN Marks
    WHERE Marks.Mark >= 60
),

```

```

GroupWithStudents AS (
    SELECT DISTINCT Students.StudentId,
                   Students.GroupId
    FROM Students
),
AllCourses AS (
    SELECT DISTINCT StudentsWithPassedCourses.CourseId
    FROM StudentsWithPassedCourses
),
AllGroups AS (
    SELECT DISTINCT GroupWithStudents.GroupId
    FROM GroupWithStudents
),
AllGroupsWithSubjects AS (
    SELECT DISTINCT AllCourses.CourseId,
                   AllGroups.GroupId
    FROM AllCourses
    CROSS JOIN AllGroups
),
tmp1 AS (
    SELECT DISTINCT AllCourses.CourseId,
                   GroupWithStudents.StudentId,
                   GroupWithStudents.GroupId
    FROM AllCourses
    CROSS JOIN GroupWithStudents
),
tmp2 AS (
    SELECT DISTINCT StudentsWithPassedCourses.CourseId,
                   GroupWithStudents.StudentId,
                   GroupWithStudents.GroupId
    FROM StudentsWithPassedCourses
    NATURAL JOIN GroupWithStudents
),
IncorrectPairs AS (
    SELECT DISTINCT Res.CourseId,
                   Res.GroupId
    FROM (SELECT DISTINCT Setminus.CourseId,
                          Setminus.StudentId,
                          Setminus.GroupId
          FROM (SELECT DISTINCT tmp1.CourseId,
                                tmp1.StudentId,
                                tmp1.GroupId
                FROM tmp1 EXCEPT ALL
                SELECT DISTINCT tmp2.CourseId,
                                tmp2.StudentId,
                                tmp2.GroupId
                FROM tmp2
            ) AS Setminus
        ) AS Res
),
CorrectPairs AS (
    SELECT DISTINCT Result.CourseId,
                   Result.GroupId
    FROM (SELECT DISTINCT AllGroupsWithSubjects.CourseId,
                          AllGroupsWithSubjects.GroupId
          FROM AllGroupsWithSubjects EXCEPT ALL

```



```

        SELECT DISTINCT IncorrectPairs.CourseId,
                        IncorrectPairs.GroupId
        FROM IncorrectPairs
    ) AS Result
)
SELECT DISTINCT CorrectPairs.CourseId,
                Courses.CourseName,
                Correctpairs.GroupId,
                Groups.GroupName
FROM CorrectPairs
    NATURAL JOIN courses
    NATURAL JOIN Groups;

```

11 Средний балл студента по идентификатору

$$avg_{Mark, \emptyset}(\sigma_{StudentId=?}(Students \bowtie Marks))$$

```

SELECT DISTINCT avg(Marks.Mark)
FROM Students
    NATURAL JOIN Marks
WHERE Students.StudentId = ?;

```

12 Средний балл студента для всех студентов

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

```

SELECT DISTINCT Students.StudentId,
                Students.StudentName,
                avg(Marks.Mark)
FROM Students
    NATURAL JOIN Marks
GROUP BY (Students.StudentId, students.StudentName);

```

13 Средний балл средних баллов студентов каждой группы

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

```

WITH AllMarks AS (
SELECT DISTINCT Students.StudentId,
                Groups.GroupId,
                Groups.GroupName,
                Marks.CourseId,
                Marks.Mark
FROM Students
    NATURAL JOIN Groups
    NATURAL JOIN Marks
),
AverageByStudents AS (
    SELECT DISTINCT AllMarks.StudentId,
                    AllMarks.GroupId,
                    AllMarks.GroupName,
                    avg(AllMarks.Mark) AS Mark
FROM AllMarks

```

```

GROUP BY (AllMarks.StudentId, AllMarks.Groupid, AllMarks.GroupName)
)
SELECT DISTINCT AverageByStudents.GroupId,
               AverageByStudents.GroupName,
               avg(AverageByStudents.Mark)
FROM AverageByStudents
GROUP BY (AverageByStudents.GroupId, AverageByStudents.GroupName);

```

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

$$StudentsTotalCourses = \rho_{count=TotalCourses}(count_{*,\{StudentId,StudentName\}}(Students \bowtie Plans))$$

$$OnlyPassedCourses = \rho_{StudentId=MarkStudentId}(\pi_{StudentId,CourseId}(\sigma_{Mark \geq 60}(Marks)))$$

$$StudentsPassedCourses = \rho_{CourseId=PassedCourses}(count_{CourseId,\{StudentId,StudentName\}}(Students \bowtie_{StudentId=MarkStudentId} OnlyPassedCourses))$$

$$\epsilon_{NonPassed=TotalCourses-PassedCourses}(StudentsTotalCourses \bowtie StudentsPassedCourses)$$

```

WITH StudentsTotalCourses AS (
  SELECT DISTINCT Students.StudentId,
                 Students.StudentName,
                 count(*) AS TotalCourses
FROM students
   NATURAL JOIN Plan
GROUP BY (Students.StudentId, Students.StudentName)
),
OnlyPassedCourses AS (
  SELECT Marks.StudentId, Marks.CourseId
FROM Marks
WHERE Marks.Mark >= 60
),
StudentsPassedCourses AS (
  SELECT DISTINCT Students.StudentId,
                 Students.StudentName,
                 count(OnlyPassedCourses.CourseId) AS PassedCourses
FROM Students
   LEFT OUTER JOIN OnlyPassedCourses
   ON OnlyPassedCourses.StudentId = Students.StudentId
GROUP BY (Students.StudentId, Students.StudentName)
)
SELECT DISTINCT StudentsTotalCourses.StudentId,
               StudentsTotalCourses.StudentName,
               StudentsTotalCourses.TotalCourses,
               StudentsPassedCourses.PassedCourses,
               StudentsTotalCourses.TotalCourses - StudentsPassedCourses.PassedCourses
               AS NotPassedCoursed
FROM StudentsTotalCourses
   NATURAL JOIN StudentsPassedCourses;

```