Домашнее задание 6

Дарья Яковлева, M3439

08.11.2016

Задание: Составить запросы в терминах исчисления кортежей и языков Datalog и SQL для базы данных «Деканат».

Переменные и таблицы

- **G :: Groups**

  $G(GID, GName)$
- **S :: Students**

  $S(SID, SName, GID)$
- **L :: Lecturers**

  $L(LID, LName)$
- **C :: Courses**

  $C(CID, CName)$
- **M :: Marks**

  $M(Mark, CID, SID)$
- **P :: Academin plan**

  $P(LID, CID, GID)$

Запросы в терминах исчисления кортежей, языка Datalog и SQL

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

   select $S.SName$

   from S

   where $\exists C\ \exists M(C.CName =$ 'Базы данных' $\wedge\ S.SID = M.SID \wedge M.CID =$ C.CID $\wedge\ M.Mark = X)$

   ————————

   $Students(SName)$:- $S(SID, SName, \_), C(CID, CName), M(Mark, CID, SID)$, $CName =$ 'Базы данных', $Mark = X$

   ————————

```
SELECT students.student_name
FROM students
WHERE
EXISTS (
SELECT *
FROM marks
WHERE (
students.student_id = marks.student_id
AND marks.mark_value = 100
AND marks.course_id in (
SELECT courses.course_id
FROM courses
WHERE courses.course_name = ’Databases’
)
)
);
```

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

   (a) **среди всех студентов**

   select $S.SName$

   from S

   where $\exists C \; \forall M \; (C.CName =$ ’Базы данных’ $\wedge \; S.SID = M.SID \wedge M.CID <> C.CID)$

   —————————

   $Students(SName)$:- $S(SID, SName, \_), C(CID, CName)$, not $M(\_, CID, SID)$, $CName =$ ’Базы данных’

   —————————

```
SELECT students.student_name
FROM students
WHERE
NOT EXISTS (
SELECT *
FROM marks
WHERE (
students.student_id = marks.student_id
AND marks.course_id in (
SELECT courses.course_id
FROM courses
WHERE courses.course_name = ’Databases’
)
)
);
```

(b) **среди студентов, у которых есть этот предмет**

select $S.SName$

from S

where $\exists P\ \exists C\ \forall M\ (S.GID = P.GID \land P.CID = C.CID \land C.CName =$ 'Базы данных' $\land\ S.SID = M.SID \land M.CID <> C.CID)$

——————————

$Students(SName)$:- $S(SID, SName, GID), C(CID, CName), P(\_, CID, GID)$, not $M(\_, CID, SID)$, $CName =$ 'Базы данных'

——————————

```
SELECT students.student_name
FROM students
WHERE
students.group_id IN (
SELECT academicplan.group_id
FROM academicplan
WHERE students.group_id = academicplan.group_id AND academicplan.course_id in (
SELECT courses.course_id
FROM courses
WHERE courses.course_name = 'Databases'
)
) AND NOT EXISTS (
SELECT *
FROM marks
WHERE (
students.student_id = marks.student_id
AND marks.course_id in (
SELECT courses.course_id
FROM courses
WHERE courses.course_name = 'Databases'
)
)
);
```

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

select $S.SName$

from S

where $\exists L\ \exists P\ \exists M\ (L.LName = X \land P.LID = L.LID \land S.SID = M.SID \land M.CID = P.CID)$

——————————

$Students(SName)$:- $S(SID, SName, \_), L(LID, LName), P(LID, CID, \_), M(\_, CID, SID)$, $LName = X$

——————————

```
SELECT students.student_name
FROM students
WHERE
students.student_id IN (
SELECT marks.student_id
FROM marks
WHERE marks.course_id IN (
SELECT academicplan.course_id
FROM academicplan
WHERE academicplan.lecturer_id IN (
SELECT lecturers.lecturer_id
FROM lecturers
WHERE lecturers.lecturer_name = 'Georgiy Korneev'
)
)
) ;
```

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

   select $S.SName$

   from S

   where $\exists L\ \forall P\ \forall M\ (L.LName = X \land P.LID = L.LID \land M.CID = P.CID \land S.SID <> M.SID)$

   ——————————

   $StudentsHaveMark(SName)$:- $S(SID, SName, \_)$, $L(LID, LName)$, $P(LID, CID, \_)$, $M(\_, CID, SID)$, $LName = X$

   $Students(SName)$:- $S(\_, SName, \_)$, not $StudentsHaveMark(SName)$

   ——————————

```
SELECT students.student_name
FROM students
EXCEPT
SELECT DISTINCT students.student_name
FROM students
WHERE
students.student_id IN (
SELECT marks.student_id
FROM marks
WHERE marks.course_id IN (
SELECT academicplan.course_id
FROM academicplan
WHERE academicplan.lecturer_id IN (
SELECT lecturers.lecturer_id
FROM lecturers
WHERE lecturers.lecturer_name = 'Georgiy Korneev'
)
)
) ;
```

5. **Всех студентов, имеющих оценки по всем предметам заданного лектора**

select $S.SName$

from S

where $\exists L\ \forall P\ \exists M\ (L.LName = X \land P.LID = L.LID \land M.CID = P.CID \land S.SID = M.SID)$

——————————

$LecturerCourses(CID)$:- $L(LID, LName)$, $P(LID, CID, \_)$, $LName = X$

$CoursesWithNoMark(SID, CID)$:- $S(SID, \_, \_), C(CID, \_)$, not $M(\_, CID, SID)$

$Students(SName)$:- $S(SID, SName, \_)$, $LecturerCourses(CID)$,

not $CoursesWithNoMark(SID, CID)$

——————————

```
SELECT students.student_name
FROM students
WHERE
students.student_id IN (
SELECT marks.student_id
FROM marks
WHERE marks.course_id IN (
SELECT academicplan.course_id
FROM academicplan
WHERE academicplan.lecturer_id IN (
SELECT lecturers.lecturer_id
FROM lecturers
WHERE lecturers.lecturer_name = 'Georgiy Korneev'
)
) AND marks.course_id NOT IN (
SELECT courses.course_id
FROM courses
WHERE courses.course_id NOT IN (
SELECT academicplan.course_id
FROM academicplan
where academicplan.group_id = students.group_id
)
)
) ;
```

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

   select $S.SName, C.CName$

   from S, C

   where $\forall P\ (S.GID = P.GID \land P.CID = C.CID)$

   ——————————

   $StudentCourses(SName, CName) \text{ :- } S(\_, SName, GID), C(CID, CName), P(\_, CID, GID)$

   ——————————

```
SELECT students.student_name, courses.course_name
FROM students, courses
WHERE students.group_id IN (
SELECT academicplan.group_id
FROM academicplan
WHERE courses.course_id = academicplan.course_id
)
ORDER BY students.student_name;
```

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

   select $S.SName$

   from S

   where $\exists L\ \exists P\ (L.LName = X \land P.LID = L.LID \land S.GID = P.GID)$

   ——————————

   $Students(SName) \text{ :- } S(\_, SName, GID), L(LID, LName), P(LID, \_, GID), LName = X$

   ——————————

```
SELECT students.student_name
FROM students
WHERE
students.group_id IN (
SELECT academicplan.group_id
FROM academicplan
WHERE academicplan.lecturer_id IN (
SELECT lecturers.lecturer_id
FROM lecturers
WHERE lecturers.lecturer_name = 'Georgiy Korneev'
)
);
```

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

   select $S.SName\ as\ S1$, $S.SName\ as\ S2$

   from S

   where $\forall M1\ \exists M2\ (M1.SID = S1.SID \land M1.Mark \geq 60 \land M2.SID = S2.SID \land M2.Mark \geq 60)$ ————— $StudentPassCourses(SName, CID)$:- $S(SID, SName, \_)$, $M(Mark, CID, SID)$, $Mark \geq 60$

   $Students(SName1, SName2)$:- $StudentPassCourses(SName1, CID)$, $S(SID, SName2, \_)$, $M(Mark, CID, SID)$, $Mark \geq 60$

   ——————

Приложение. База данных.

```
CREATE TABLE IF NOT EXISTS groups (
group_id int PRIMARY KEY,
group_name varchar(100)
);

CREATE TABLE IF NOT EXISTS students (
student_id int PRIMARY KEY,
student_name varchar(100),
group_id int REFERENCES groups (group_id)
);

CREATE TABLE IF NOT EXISTS lecturers (
lecturer_id int PRIMARY KEY,
lecturer_name varchar(50)
);


CREATE TABLE IF NOT EXISTS courses (
course_id int PRIMARY KEY,
course_name varchar(50)
);

CREATE TABLE IF NOT EXISTS marks (
mark_value int,
course_id int not null REFERENCES courses (course_id),
student_id int not null REFERENCES students (student_id),
CHECK (mark_value BETWEEN 0 AND 100),
PRIMARY KEY (course_id, student_id)
);

CREATE TABLE IF NOT EXISTS academicplan (
lecturer_id int not null REFERENCES lecturers (lecturer_id),
course_id int not null REFERENCES courses (course_id),
group_id int not null REFERENCES groups (group_id),
PRIMARY KEY (lecturer_id, course_id, group_id)
);
```