Кацман Алексей
M3438

1 Заполнение тестовыми данными в SQL

Листинг 1: Заполнение тестовыми данными

```
drop database if exists deanery_6;
create database deanery_6;
use deanery_6;

create table Students(
    s_id int primary key,
    s_name varchar(50)) default charset='utf8';

create table Groups(
    g_id int primary key,
    g_name varchar(5)) default charset='utf8';

create table Lecturers(
    l_id int primary key,
    l_name varchar(50)) default charset='utf8';

create table Courses(
    c_id int primary key,
    c_name varchar(50)) default charset='utf8';

create table StudentGroups(
    s_id int primary key,
    g_id int) default charset='utf8';

create table Marks(
    s_id int,
    c_id int,
    mark char,
    primary key (s_id, c_id)) default charset='utf8';

create table Plans(
    g_id int,
    c_id int,
    l_id int,
    primary key (g_id, c_id)) default charset='utf8';
```

```
36
37 insert into Students (s_id, s_name) values
38     (1, "Ivanov"),
39     (2, "Petrov"),
40     (3, "Sidorov"),
41     (4, "Fedorov"),
42     (5, "Timofeev");
43
44 insert into Groups (g_id, g_name) values
45     (1, "M3437"),
46     (2, "M3438"),
47     (3, "M3439");
48
49 insert into Courses (c_id, c_name) values
50     (1, "Databases"),
51     (2, "Java"),
52     (3, "Algorithms");
53
54 insert into Lecturers (l_id, l_name) values
55     (1, "Korneev"),
56     (2, "Stankevich");
57
58 insert into StudentGroups (s_id, g_id) values
59     (1, 1),
60     (2, 2),
61     (3, 2),
62     (4, 3),
63     (5, 3);
64
65 insert into Plans (g_id, c_id, l_id) values
66     (1, 1, 1),
67     (1, 2, 1),
68     (1, 3, 2),
69     (2, 1, 1),
70     (2, 2, 1),
71     (2, 3, 2),
72     (3, 2, 1);
73
74 insert into Marks (s_id, c_id, mark) values
75     (1, 1, 'B'),
76     (1, 2, 'A'),
77     (1, 3, 'C'),
78     (2, 1, 'B'),
```

```
79      (3, 1, 'D');
```

2 ДЗ 6

2.1 Информация о студентах с заданной оценкой по предмету "Базы данных"

- Исчисление кортежей

  $S :: Students; C :: Courses; M :: Marks;$ select S where $\exists C$ ($C.c\_name$ = "Databases"$\wedge$ $\exists M$ ($s\_id = M.s\_id \wedge M.mark =$ MARK $\wedge$ $C.c\_id = M.c\_id$))

  MARK - заданная оценка

- Datalog

  $A(s\_id)$:- $Marks(s\_id,$ Course, MARK), $Courses($Course, "Databases")

  MARK - заданная оценка

- SQL

Листинг 2: Задание 1

```
1 select s_id from Students, Courses where
2     Courses.c_name = "Databases" and
3     (exists (select * from Marks where
4             Students.s_id = Marks.s_id and
5             Marks.mark = 'B' and
6             Courses.c_id = Marks.c_id));
```

2.2 Информация о студентах не имеющих оценки по предмету "Базы данных"

2.2.1 среди всех студентов

- Исчисление кортежей

  $S :: Students; C :: Courses; M :: Marks;$ select S where $!\exists C$ ($C.c\_name$ = "Databases"$\wedge$ $\exists M$ ($s\_id = M.s\_id \wedge C.c\_id = M.c\_id$))

- Datalog

  $ExistingMarks$('A'); $ExistingMarks$('B'); $ExistingMarks$('C'); $ExistingMarks$('D'); $ExistingMarks$('E')

  $A(s\_id)$:- $Students(s\_id, \_)$, not $Marks(s\_id, \text{Course}, \text{Mark})$, $Courses(\text{Course}, \text{"Databases"})$, $ExistingMarks(\text{Mark})$

- SQL

Листинг 3: Задание 2(а)

```
select s_id from Students where
    not exists (select * from Courses where
        Courses.c_name = "Databases" and
        (exists (select * from Marks where
                Students.s_id = Marks.s_id and
                Courses.c_id = Marks.c_id)));
```

2.2.2 среди студентов, у которых есть этот предмет

- Исчисление кортежей

  $S :: Students$; $C :: Courses$; $M :: Marks$; $SG :: StudentGroups$; $P :: Plans$; select S where $!\exists C\ (C.c\_name = \text{"Databases"} \wedge \exists M\ (s\_id = M.s\_id \wedge C.c\_id = M.c\_id)) \wedge \exists SG,\ P\ (s\_id = SG.s\_id \wedge SG.s\_id = P.s\_id \wedge C.c\_id = P.c\_id)$

- Datalog

  $ExistingMarks$('A'); $ExistingMarks$('B'); $ExistingMarks$('C'); $ExistingMarks$('D'); $ExistingMarks$('E')

  $A(s\_id)$:- $StudentGroups(s\_id, g\_id)$, $Plans(g\_id, \text{Course}, \_)$, $Courses(\text{Course}, \text{"Databases"})$

  $B(s\_id)$:- $Students(s\_id, \_)$, not $Marks(s\_id, \text{Course}, \text{Mark})$, $Courses(\text{Course}, \text{"Databases"})$, $ExistingMarks(\text{Mark})$

  $C(s\_id)$:- $A(s\_id)$, $B(s\_id)$

- SQL

Листинг 4: Задание 2(б)

```
select s_id from Students where
    not exists (select * from Courses where
        Courses.c_name = "Databases" and
        (exists (select * from Marks where
```

```
5                   Students.s_id = Marks.s_id and
6                   Courses.c_id = Marks.c_id)) and
7           (exists (select * from StudentGroups where
8                   exists (select * from Plans where
9                       StudentGroups.s_id = Students.s_id
                           and
10                      StudentGroups.g_id = Plans.g_id and
11                      Plans.c_id = Courses.c_id))));
```

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

- Исчисление кортежей

  $S :: Students$; $M :: Marks$; $L :: Lecturers$; $SG :: StudentGroups$; $P :: Plans$; select S where $\exists M,\ L,\ SG,\ P\ (s\_id = M.s\_id \wedge P.s\_id = M.s\_id \wedge s\_id = SG.s\_id \wedge P.s\_id = SG.s\_id \wedge L.l\_id = P.l\_id \wedge L.l\_name =$ LNAME

  LNAME - заданное имя лектора

- Datalog

  $A(s\_id)$:- $Marks(s\_id, g\_id, \_), StudentGroups(s\_id, g\_id), Plans(g\_id, c\_id, l\_id)$, $Lecturers(l\_id,$ LNAME)

  LNAME - заданное имя лектора

- SQL

Листинг 5: Задание 3

```
1 select distinct Students.s_id from Students, Lecturers,
      Marks, StudentGroups, Plans where
2     Marks.s_id = Students.s_id and
3     Marks.c_id = Plans.c_id and
4     StudentGroups.s_id = Students.s_id and
5     StudentGroups.g_id = Plans.g_id and
6     Lecturers.l_id = Plans.l_id and
7     Lecturers.l_name = "Korneev";
```

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

- Исчисление кортежей

$S :: Students$; $M :: Marks$; $L :: Lecturers$; $SG :: StudentGroups$; $P :: Plans$; select S where $!\exists M,\ L,\ SG,\ P\ (s\_id = M.s\_id \wedge P.s\_id = M.s\_id \wedge s\_id = SG.s\_id \wedge P.s\_id = SG.s\_id \wedge L.l\_id = P.l\_id \wedge L.l\_name =$ LNAME

LNAME - заданное имя лектора

- Datalog

  $A(s\_id)$:- $Marks(s\_id, g\_id, \_), StudentGroups(s\_id, g\_id), Plans(g\_id, c\_id, l\_id), Lecturers(l\_id,$ LNAME)

  $B(s\_id)$:- $Students(s\_id,\ \_)$, not $A(s\_id)$

  LNAME - заданное имя лектора

- SQL

Листинг 6: Задание 4

```
select Students.s_id from Students where
    not exists (select * from Lecturers, Marks,
        StudentGroups, Plans where
         Marks.s_id = Students.s_id and
         Marks.c_id = Plans.c_id and
         StudentGroups.s_id = Students.s_id and
         StudentGroups.g_id = Plans.g_id and
         Lecturers.l_id = Plans.l_id and
         Lecturers.l_name = "Korneev");
```

2.5 Все студенты, имеющие оценки по всем предметам заданного лектора

- Исчисление кортежей

  $S :: Students$; $M :: Marks$; $L :: Lecturers$; $SG :: StudentGroups$; $P :: Plans$; select S where $\exists L\ (L.l\_name =$ LNAME $\wedge\ \forall M,\ SG,\ P\ ((s\_id = SG.s\_id \wedge P.s\_id = SG.s\_id \wedge L.l\_id = P.l\_id) \rightarrow (s\_id = M.s\_id \wedge P.c\_id = M.c\_id)))$

  LNAME - заданное имя лектора

- Datalog

  $ExistingMarks$('A'); $ExistingMarks$('B'); $ExistingMarks$('C'); $ExistingMarks$('D'); $ExistingMarks$('E')

$A(s\_id)$:- $Students(s\_id, \_)$, $StudentGroups(s\_id, g\_id)$, $Lecturers(l\_id,$ LNAME), $Plans(g\_id,\ c\_id,\ l\_id)$, not $Marks(s\_id,\ c\_id,$ Mark), $ExistingMarks$(Mark)

$B(s\_id)$:- $Students(s\_id,\ \_)$, not $A(s\_id)$

LNAME - заданное имя лектора

- SQL

Листинг 7: Задание 5

```
select Students.s_id from Students, Lecturers where
    Lecturers.l_name = "Stankevich" and
    not exists (select * from Marks, StudentGroups,
        Plans where
        not (not (StudentGroups.s_id = Students.s_id
            and
                StudentGroups.g_id = Plans.g_id and
                Lecturers.l_id = Plans.l_id) or
            (Marks.c_id = Plans.c_id and
                Marks.s_id = Students.s_id)));
```

2.6 Для каждого студента имя и курсы, которые он должен посещать

- Исчисление кортежей

  $S :: Students$; $C :: Courses$; $SG :: StudentGroups$; $P :: Plans$; select $S.s\_name$, $C.c\_name$ from S, C where $\exists S,\ C, SG,\ P$ $(S.s\_id = SG.s\_id \wedge P.g\_id = SG.g\_id \wedge C.c\_id = P.c\_id)$

- Datalog

  $A(s\_name, c\_name)$:- $Students(s\_id, s\_name)$, $StudentGroups(s\_id,$ $g\_id)$, $Plans(g\_id,\ c\_id,\ \_)$, $Courses(c\_id,\ c\_name)$

- SQL

Листинг 8: Задание 6

```
select Students.s_name, Courses.c_name from Students,
    Courses, StudentGroups, Plans where
    Students.s_id = StudentGroups.s_id and
    StudentGroups.g_id = Plans.g_id and
    Plans.c_id = Courses.c_id;
```

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

- Исчисление кортежей

  $S :: Students$; $L :: Lecturers$; $SG :: StudentGroups$; $P :: Plans$; select $L.l\_name$, $S.s\_name$ from L, S where $\exists S$, L, SG, P $(S.s\_id = SG.s\_id \wedge P.g\_id = SG.g\_id \wedge L.l\_id = P.l\_id)$

- Datalog

  $A(l\_name, s\_name)$:- $Students(s\_id, s\_name)$, $StudentGroups(s\_id, g\_id)$, $Plans(g\_id, \_, l\_id)$, $Lecturers(l\_id, l\_name)$

- SQL

Листинг 9: Задание 7

```
1 select distinct Lecturers.l_name, Students.s_name from
    Students, Lecturers, StudentGroups, Plans where
2    Students.s_id = StudentGroups.s_id and
3    StudentGroups.g_id = Plans.g_id and
4    Plans.l_id = Lecturers.l_id;
```

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

- Исчисление кортежей

  $S1 :: Students$; $S2 :: Students$; $M1 :: Marks$, $M2 :: Marks$; select $S1.s\_id$, $S2.s\_id$ from $S1$, $S2$ where $\forall M1$ $(S1.s\_id = M1.s\_id \rightarrow \exists M2$ $(S2.s\_id = M2.s\_id \wedge M1.c\_id = M2.c\_id))$

- Datalog

  $ExistingMarks$('A'); $ExistingMarks$('B'); $ExistingMarks$('C'); $ExistingMarks$('D'); $ExistingMarks$('E')

  $A(s1\_id, s2\_id)$:- $Students(s1\_id, \_)$, $Students(s2\_id, \_)$

  $B(s1\_id, s2\_id)$:- $Students(s1\_id, \_)$, $Students(s2\_id, \_)$, $Marks(s1\_id, c\_id, \text{Mark1})$, not $Marks(s2\_id, c\_id, \text{Mark2})$, $ExistingMarks(\text{Mark1})$, $ExistingMarks(\text{Mark2})$

  $C(s1\_id, s2\_id)$:- $A(s1\_id, s2\_id)$, not $B(s1\_id, s2\_id)$

- SQL

Листинг 10: Задание 8

```
select S1.s_id, S2.s_id
    from Students as S1, Students as S2 where
        not exists (select * from Marks as M1 where
            not (not S1.s_id = M1.s_id or
                (exists (select * from Marks as M2
                    where
                        S2.s_id = M2.s_id and M1.c_id =
                            M2.c_id))));
```