

```

1  -- Schema (without constraints):
2
3  --
4  -- CREATE TABLE Student (
5  --     Id SERIAL NOT NULL PRIMARY KEY,
6  --     Name VARCHAR (50) NOT NULL,
7  --     GroupId INT NOT NULL
8  -- );
9  -- CREATE TABLE Lecturer (
10 --     Id SERIAL NOT NULL PRIMARY KEY,
11 --     Name VARCHAR (50) NOT NULL
12 -- );
13 -- CREATE TABLE StudentGroup (
14 --     Id SERIAL NOT NULL PRIMARY KEY,
15 --     Name VARCHAR (50) NOT NULL UNIQUE
16 -- );
17 -- CREATE TABLE Course (
18 --     Id SERIAL NOT NULL PRIMARY KEY,
19 --     Name VARCHAR (50) NOT NULL
20 -- );
21 -- CREATE TABLE Teaching (
22 --     GroupId INT NOT NULL,
23 --     CourseId INT NOT NULL,
24 --     LecturerId INT NOT NULL,
25 --     PRIMARY KEY (GroupId, CourseId)
26 -- );
27 -- CREATE TABLE Mark (
28 --     StudentId INT NOT NULL,
29 --     CourseId INT NOT NULL,
30 --     Mark DECIMAL (4, 2) NOT NULL,
31 --     PRIMARY KEY (StudentId, CourseId)
32 -- );
33
34
35 -----
36 -- #1
37 -- Информацию о студентах с заданной оценкой по предмету «Базы данных» .
38 -----
39
40 -- Let DBMS course have id = 1, interesting mark = 67
41
42 SELECT s.* FROM Student s
43 INNER JOIN Teaching t ON t.GroupId = s.GroupId
44 INNER JOIN Mark m ON s.Id = m.StudentId AND t.CourseId = m.CourseId
45 WHERE m.CourseId = 1 AND m.Mark = 67
46 ;
47
48 -----
49 -- #2a
50 -- Информацию о студентах не имеющих оценки по предмету «Базы данных» :
51 --     (1) среди всех студентов
52 -----
53
54 SELECT s.* FROM Student s
55 LEFT JOIN Mark m ON s.Id = m.StudentId AND m.CourseId = 1
56 WHERE m.Mark IS NULL
57 ;
58
59 -----
60 -- #2b
61 -- Информацию о студентах не имеющих оценки по предмету «Базы данных» :
62 --     (2) среди студентов , у которых есть этот предмет

```

```

63  -- -----
64
65  SELECT s.* FROM Student s
66  INNER JOIN Teaching t ON t.GroupId = s.GroupId
67  LEFT JOIN Mark m ON s.Id = m.StudentId AND m.CourseId = t.CourseId
68  WHERE m.Mark IS NULL AND t.CourseId = 1
69  ;
70
71  -- -----
72  -- #3
73  -- Информацию о студентах , имеющих хотя бы одну оценку у заданного лектора .
74  -- -----
75
76  SELECT s.* FROM Student s
77  INNER JOIN Teaching t ON t.GroupId = s.GroupId
78  INNER JOIN Mark m ON s.Id = m.StudentId AND t.CourseId = m.CourseId
79  WHERE t.LecturerId = 1
80  GROUP BY s.id, t.LecturerId
81  ;
82
83  -- -----
84  -- #4
85  -- Идентификаторы студентов , не имеющих ни одной оценки у заданного лектора
86  -- -----
87
88  SELECT s.* FROM Student s
89  LEFT JOIN Teaching t ON t.GroupId = s.GroupId AND t.LecturerId = 1
90  LEFT JOIN Mark m ON s.Id = m.StudentId AND t.CourseId = m.CourseId
91  WHERE m.Mark IS NULL
92  ;
93
94  -- -----
95  -- #5a
96  -- Студентов , имеющих оценки по всем предметам заданного лектора .
97  -- (имеющие оценки по всем предметам заданного лектора , которые они должны
98  -- посещать )
99  -- -----
100 SELECT s.* FROM Teaching t
101 INNER JOIN Student s ON s.GroupId = t.GroupId
102 LEFT JOIN Mark m ON m.CourseId = t.CourseId AND s.Id = m.StudentId
103 WHERE t.LecturerId = 1
104 GROUP BY s.Id
105 HAVING bool_and(m.Mark IS NOT NULL)
106 ;
107
108 -- -----
109 -- #5b
110 -- Студентов , имеющих оценки по всем предметам заданного лектора .
111 -- (имеющие оценки по всем предметам , которые читает заданный лектор )
112 -- -----
113
114 SELECT s.* FROM Teaching t
115 CROSS JOIN Student s
116 LEFT JOIN Mark m ON m.CourseId = t.CourseId AND s.Id = m.StudentId
117 WHERE t.LecturerId = 1
118 GROUP BY s.Id
119 HAVING bool_and(m.Mark IS NOT NULL)
120 ;
121
122 -- -----

```

```

123 -- #6
124 -- Для каждого студента имя и предметы , которые он должен посещать .
125 -- -----
126
127 SELECT s.Id, s.Name, c.Id, c.Name FROM Student s
128 INNER JOIN Teaching t ON t.GroupId = s.GroupId
129 INNER JOIN Course c ON c.Id = t.CourseId
130 ;
131
132 -- -----
133 -- #7
134 -- По лектору всех студентов , у которых он хоть что-нибудь преподавал .
135 -- -----
136
137 SELECT s.* FROM Teaching t
138 INNER JOIN Student s ON s.GroupId = t.GroupId
139 WHERE t.LecturerId = 1
140 ;
141
142 -- -----
143 -- #8
144 -- Пары студентов , такие , что все сданные первым студентом предметы сдал и
145 -- второй студент .
146 -- -----
147
148 SELECT s1.Id, s1.Name, s2.Id, s2.Name FROM Mark m1
149 INNER JOIN Student s1 ON s1.Id = m1.StudentId
150 INNER JOIN Student s2 ON s2.Id > m1.StudentId
151 LEFT JOIN Mark m2 ON m1.CourseId = m2.CourseId AND m2.StudentId = s2.Id
152 GROUP BY s1.Id, s2.Id
153 HAVING bool_and(m2.Mark IS NOT NULL)
154 ;
155
156 -- -----
157 -- #9
158 -- Такие группы и предметы , что все студенты группы сдали предмет .
159 -- -----
160
161 SELECT g.Name, c.Id, c.Name FROM Teaching t
162 INNER JOIN StudentGroup g ON g.Id = t.GroupId
163 INNER JOIN Course c ON c.Id = t.CourseId
164 INNER JOIN Student s ON s.GroupId = g.Id
165 LEFT JOIN Mark m ON m.StudentId = s.Id AND m.CourseId = c.Id
166 GROUP BY g.Id, c.Id
167 HAVING bool_and(m.Mark IS NOT NULL)
168 ;
169
170 -- -----
171 -- #10a
172 -- Средний балл студента .
173 -- по идентификатору
174 -- (учитываются только предметы , по которым у студента есть оценка )
175 -- -----
176
177 SELECT avg(m.Mark) FROM Mark m
178 WHERE m.StudentId = 1
179 ;
180
181 -- -----
182 -- #10b
183 -- Средний балл студента .
184 -- для каждого студента

```

```
184 -- (учитываются только предметы , по которым у студента есть оценка )
185 -- -----
186
187 SELECT s.*, avg(m.Mark) FROM Mark m
188 JOIN Student s ON s.Id = m.StudentId
189 GROUP BY s.Id
190 ;
191
192 -- -----
193 -- #11
194 -- Средний балл средних баллов студентов каждой группы .
195 -- (учитываются только предметы , по которым у студента есть оценка )
196 -- -----
197
198 SELECT g.*, avg(m.mark) FROM StudentGroup g
199 JOIN Student s ON s.GroupId = g.Id
200 JOIN (SELECT s.Id, avg(m.Mark) as mark FROM Mark m
201       JOIN Student s ON s.Id = m.StudentId
202       GROUP BY s.Id) m ON m.Id = s.Id
203 GROUP BY g.Id
204 ;
205
206 -- -----
207 -- #12
208 -- Для каждого студента число предметов , которые у него были ,
209 -- число сданных предметов и число не сданных предметов .
210 -- -----
211
212 SELECT s.*, t1.c as total, t2.c as closed, t1.c - t2.c as open FROM Student
213 s
214 JOIN (SELECT count(*) as c, t.GroupId as gid FROM Teaching t GROUP BY t.GroupId) as t1 ON t1.gid = s.GroupId
215 JOIN (SELECT count(m.Mark) as c, s.Id as sid FROM Teaching t
216       JOIN Student s ON t.GroupId = s.GroupId
217       LEFT JOIN Mark m ON s.Id = m.StudentId AND m.CourseId = t.CourseId
218       GROUP BY s.Id) as t2 ON t2.sid = s.Id
```