ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | . |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6 |
| Разработка SQL запросов: запросы с подзапросами |
| по дисциплине: ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4131 |  | 29.11.2023 |  | В.А.Алексеев |
|  |  |  | подпись, дата |  | инициалы, фамилия |

1. **Задание.**

Реализовать запросы г) .. ж), указанные в варианте задания.

Один из запросы на максимум/минимум реализовать и с помощью директивы all(где возможно) и с помощью агрегатных функций.

Запрос на «все» (реляционное деление) реализовать с помощью 2 not exists и с помощью агрегатной функции.

Запросы на разность реализовать в 3 вариантах: Not in, except (MySQL не поддерживает, поэтому только синтаксис), с использованием левого/правого соединения.

Использовать with можно не более чем в1 запросе.

Программа для рисования графов: название вершины, координаты левой верхней точки отображаемой вершины, и еѐ размеры, автор графа, пользователи, которым можно просматривать или редактировать граф

а. Вершины, название/текст которых содержит слово «стоп», но не начинается с него

б. Вершины, у которых нет исходящих ребер

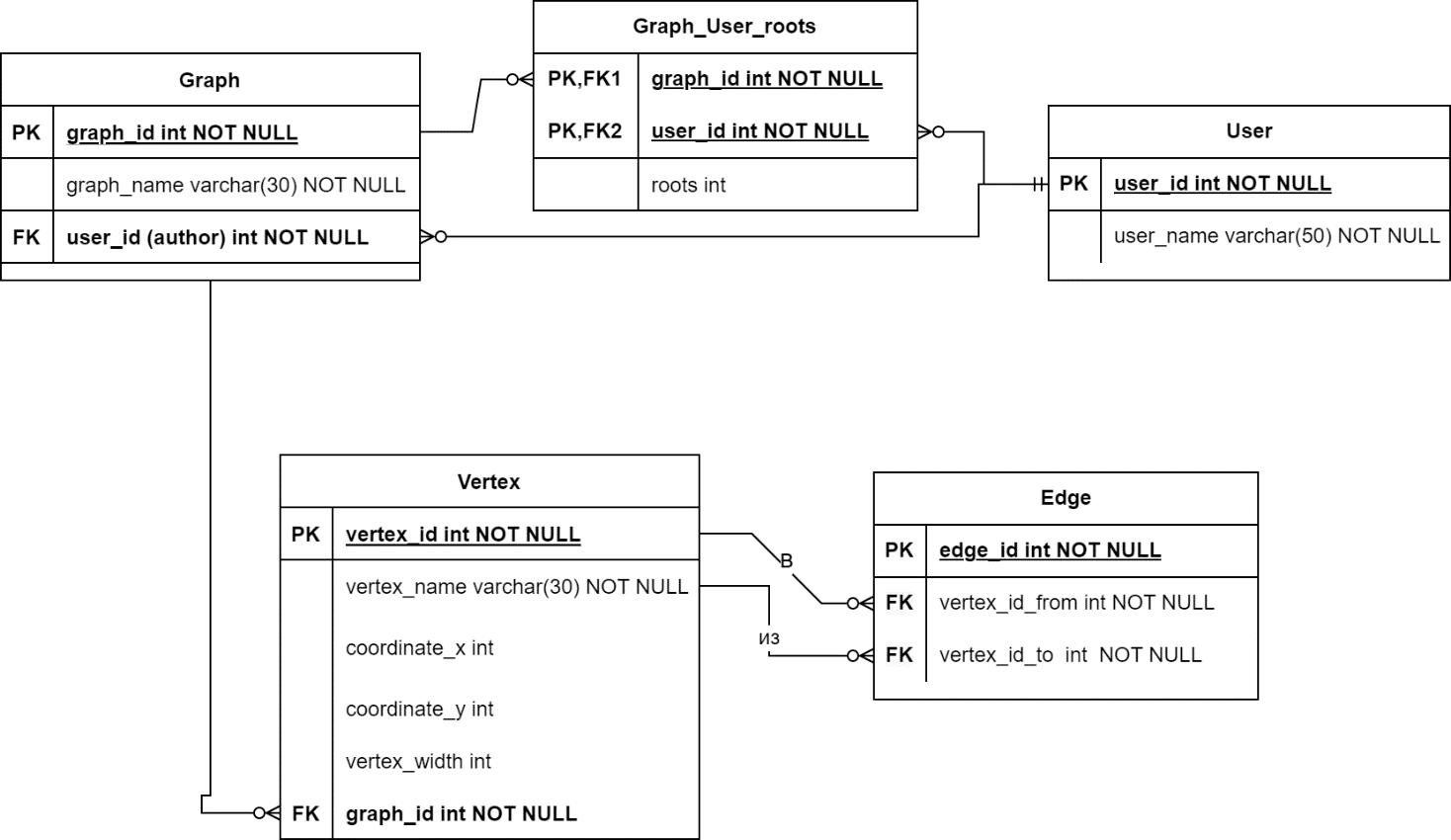
в. Графы, в которых есть вершина с петлей.

г. Графы, ширина которых не превышает 300 пикселей.Ширина графа в пикселях (от максимальной сумма координаты по горизонтали с шириной отнять минимальную левую координату)

д. авторы графов, графы которых может редактировать наибольшее число пользователей

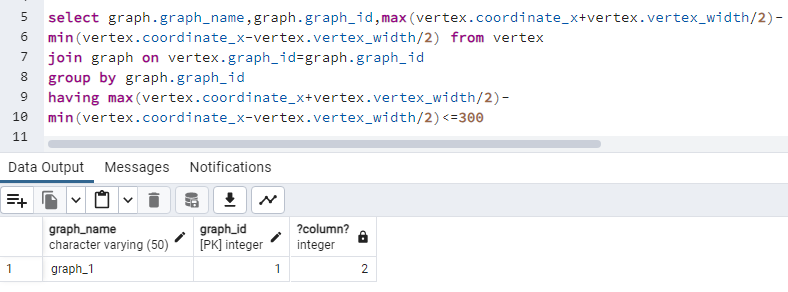
е. Вершины, для которых есть исходящие ребра, ведущие ко всем остальным вершинам еѐ графа ж. Вершины, у которых нет входящих ребер от вершины со словом «выход»

1. **Физическая модель БД.**

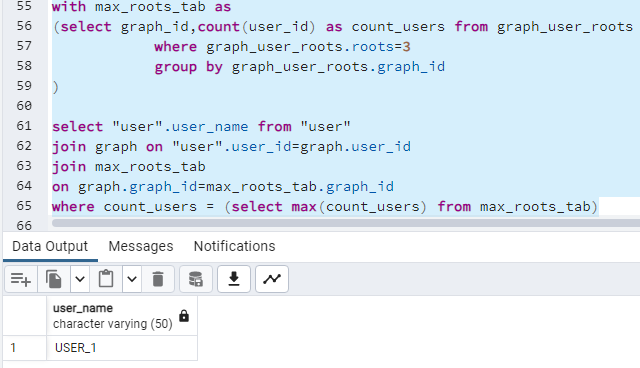


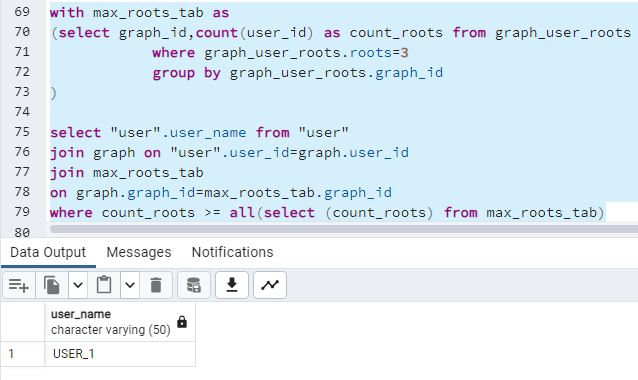
1. **Наборы данных, возвращаемые запросами.**

**Г)**

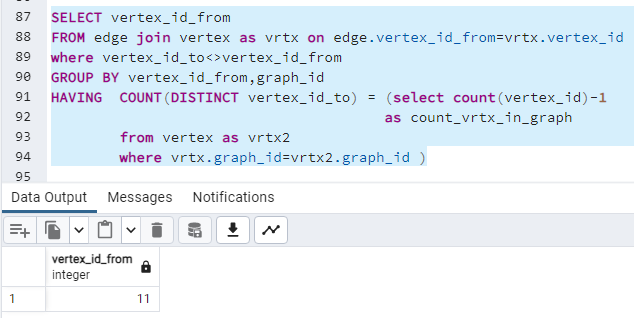


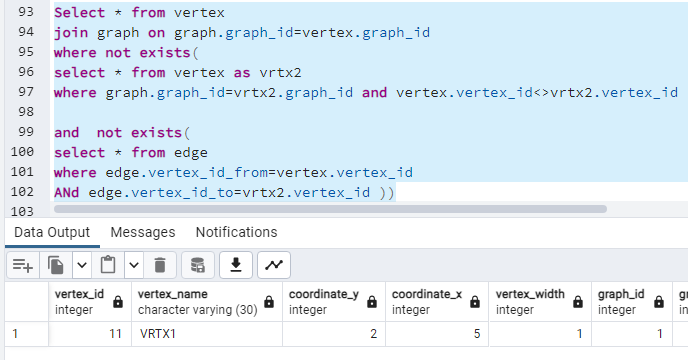
Д)



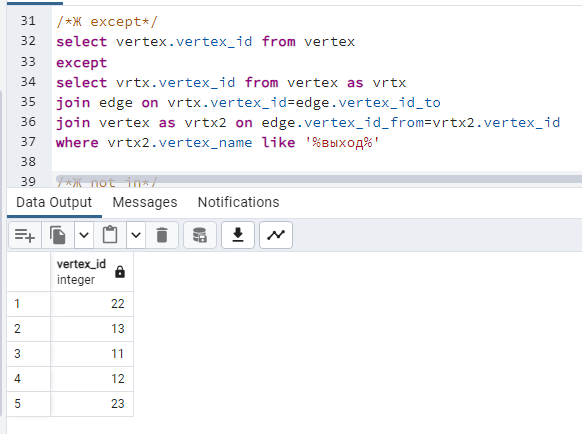


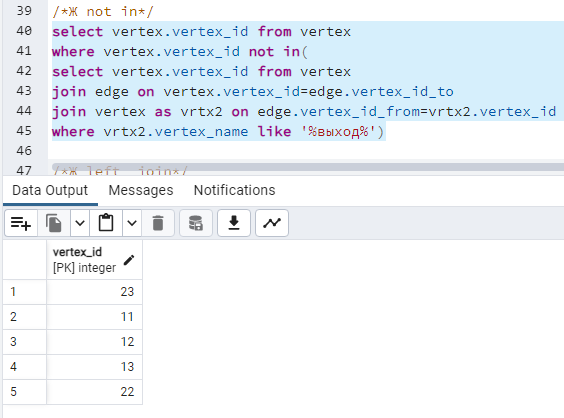
Е)

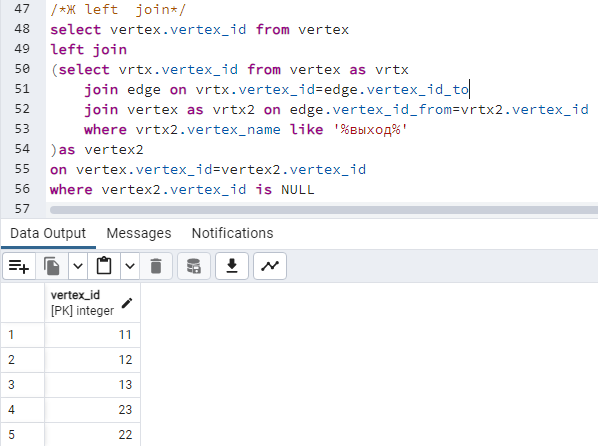




Ж)







1. **Таблица тестовых данных.**

|  |  |  |
| --- | --- | --- |
| Текст запроса | данные удовлетворяющие условиям запросов | данные не удовлетворяющие условиям запросов |
| Вершины, название/текст которых содержит слово «стоп», но не начинается с него | vertex (vertex\_name)  id=23 VRTX стоп | vertex (vertex\_name)  VRTX1 id=11,21  стоп VRTX2 id=12,22  VRTX4 id=13  выход id=24 |
| Вершины, у которых нет исходящих ребер | Vertex(vertex\_id)  13  22  23 | Vertex(vertex\_id)  11  12  21,24 |
| Графы,в которых есть вершина с петлей. | graph(name\_graph)  graph\_1 | graph(name\_graph)  graph\_2 |
| Графы, ширина которых не превышает 300 пикселей. Ширина графа в пикселях | graph(name\_graph)  graph\_1 | graph(name\_graph)  graph\_2 |
| авторы графов, графы которых может редактировать наибольшее число пользователей | USER\_1 | USER\_2 |
| Вершины, для которых есть исходящие ребра, ведущие ко всем остальным вершинам еѐ графа | Vertex(vertex\_id)  11 | Vertex(vertex\_id)  12  13  21  22  23  24 |
| Вершины, у которых нет входящих ребер от вершины со словом «выход» | Vertex(vertex\_id)  11,12,13  22  23 | Vertex(vertex\_id)  24  21 |

1. **Текст запросов.**

/\*Графы, ширина которых не превышает 300 пикселей. Ширина графа в пикселях

Ответ:graph\_1

\*/

select graph.graph\_name,graph.graph\_id,max(vertex.coordinate\_x+vertex.vertex\_width/2)-

min(vertex.coordinate\_x-vertex.vertex\_width/2) from vertex

join graph on vertex.graph\_id=graph.graph\_id

group by graph.graph\_id

having max(vertex.coordinate\_x+vertex.vertex\_width/2)-

min(vertex.coordinate\_x-vertex.vertex\_width/2)<=300

/\*Ж

Вершины, у которых нет входящих ребер от вершины со словом «выход»

Ответ:11,12,13,22,23

\*/

/\*Ж except\*/

select vertex.vertex\_id from vertex

except

select vrtx.vertex\_id from vertex as vrtx

join edge on vrtx.vertex\_id=edge.vertex\_id\_to

join vertex as vrtx2 on edge.vertex\_id\_from=vrtx2.vertex\_id

where vrtx2.vertex\_name like '%выход%'

/\*Ж not in\*/

select vertex.vertex\_id from vertex

where vertex.vertex\_id not in(

select vertex.vertex\_id from vertex

join edge on vertex.vertex\_id=edge.vertex\_id\_to

join vertex as vrtx2 on edge.vertex\_id\_from=vrtx2.vertex\_id

where vrtx2.vertex\_name like '%выход%')

/\*Ж left join\*/

select vertex.vertex\_id from vertex

left join

(select vrtx.vertex\_id from vertex as vrtx

join edge on vrtx.vertex\_id=edge.vertex\_id\_to

join vertex as vrtx2 on edge.vertex\_id\_from=vrtx2.vertex\_id

where vrtx2.vertex\_name like '%выход%'

)as vertex2

on vertex.vertex\_id=vertex2.vertex\_id

where vertex2.vertex\_id is NULL

/\*д. авторы графов, графы которых может редактировать

наибольшее число пользователей

Ответ:USER\_1

\*/

--временные результаты запроса.

with max\_roots\_tab as

(select graph\_id,count(user\_id) as count\_users from graph\_user\_roots

where graph\_user\_roots.roots=3

group by graph\_user\_roots.graph\_id

)

select "user".user\_name from "user"

join graph on "user".user\_id=graph.user\_id

join max\_roots\_tab

on graph.graph\_id=max\_roots\_tab.graph\_id

where count\_users = (select max(count\_users) from max\_roots\_tab)

with max\_roots\_tab as

(select graph\_id,count(user\_id) as count\_roots from graph\_user\_roots

where graph\_user\_roots.roots=3

group by graph\_user\_roots.graph\_id

)

select "user".user\_name from "user"

join graph on "user".user\_id=graph.user\_id

join max\_roots\_tab

on graph.graph\_id=max\_roots\_tab.graph\_id

where count\_roots >= all(select (count\_roots) from max\_roots\_tab)

/\*Вершины, для которых есть исходящие ребра,

ведущие ко всем остальным вершинам еѐ графа\*/

SELECT vertex\_id\_from

FROM edge join vertex as vrtx on edge.vertex\_id\_from=vrtx.vertex\_id

where vertex\_id\_to<>vertex\_id\_from

GROUP BY vertex\_id\_from,graph\_id

HAVING COUNT(DISTINCT vertex\_id\_to) = (select count(vertex\_id)-1

as count\_vrtx\_in\_graph

from vertex as vrtx2

where vrtx.graph\_id=vrtx2.graph\_id )

Select \* from vertex

join graph on graph.graph\_id=vertex.graph\_id

where not exists(

select \* from vertex as vrtx2

where graph.graph\_id=vrtx2.graph\_id and vertex.vertex\_id<>vrtx2.vertex\_id

and not exists(

select \* from edge

where edge.vertex\_id\_from=vertex.vertex\_id

ANd edge.vertex\_id\_to=vrtx2.vertex\_id ))