МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБО6РОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| старший преподаватель |  |  |  | Н.В. Путилова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «Разработка SQL запросов: запросы с подзапросами» |
| по курсу: Проектирование баз данных |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Самарин Д.В |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**1. Цель работы:**

По аналогии с примерами, приведенными в п. 1 реализовать запросы г) .. ж), указанные в варианте задания. Один из запросов на максимум/минимум реализовать с помощью директивы all Запрос на «все» (реляционное деление) реализовать с помощью 2 not exists Запросы на разность реализовать в 3 вариантах: Not in,except(MySQL не поддерживает, поэтому только синтаксис), с использованием левого/правого соединения.

**2. Вариант задания (16 вариант)**

Школьные экскурсии: тип (развлекательная/образовательная), дисциплины к которым имеет отношение образовательная экскурсия, стоимость с человека, список участников, ответственный за проведение учитель

а. экскурсии на верфи (слово «верфь» в любом месте названия)

б. экскурсии, относящиеся к биологии и географии

в. учителя, не отвечающие за экскурсии

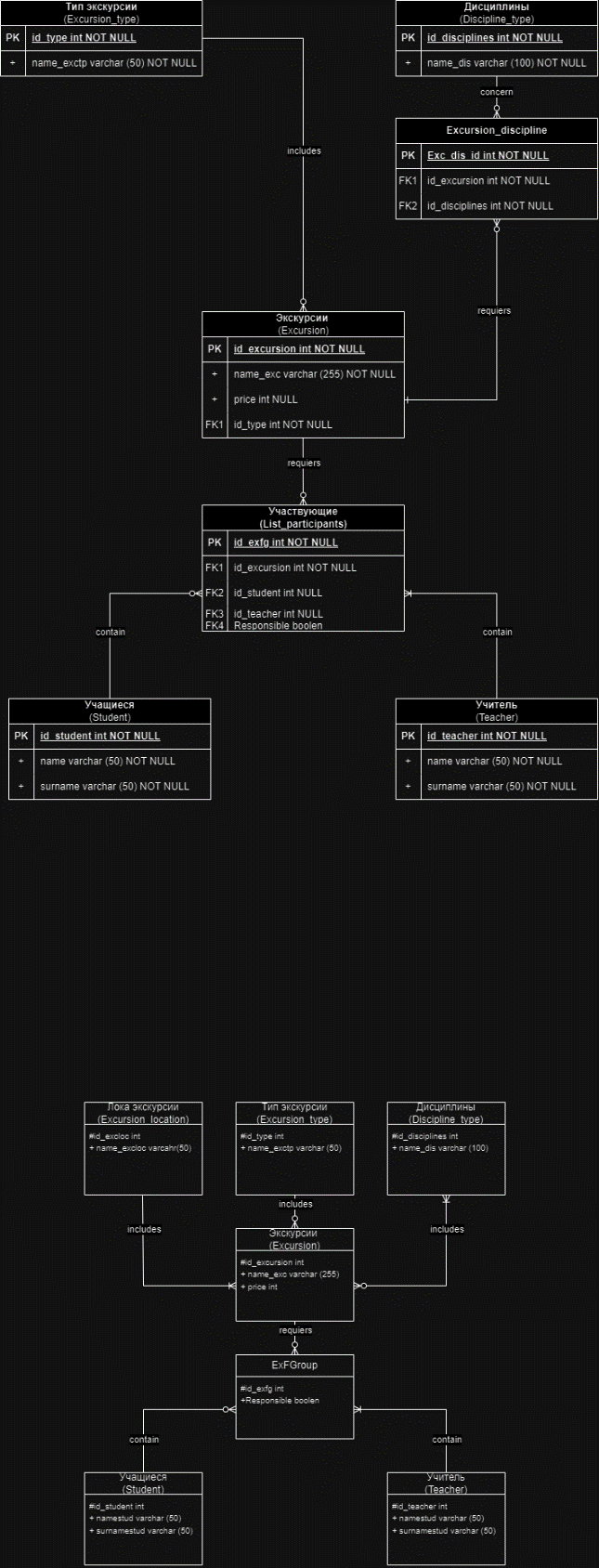
г. учащиеся, которые не ездили в музей истории религии, но ездили в Эрмитаж

д. учитель, отвечавший за число экскурсий больше среднего

е. учителя, отвечающие за самые дорогие экскурсии

ж. экскурсии, на которые ездили все учащиеся

**3. Физическая модель БД**



**4. Таблица тестовых данных:**

|  |  |  |
| --- | --- | --- |
| **Текст запроса** | **Данные, удовлетворяющие условиям запросов** | **Данные, не удовлетворяющие условиям запросов** |
| г.  учащиеся, которые не ездили в музей истории религии, но ездили в Эрмитаж | Student (Ученик)  Артем Кузнецов  Даниил Попов  Иван Новиков  Егор Петров  Анастасия Смирнова  Ольга Иванова | Student (Ученик)  Юлия Артемьева  .  .  . остальные ученики  .  . |
| д.  учитель, отвечавший за число экскурсий больше среднего | Teacher (Учитель)  Елена Петрова | Teacher (Учитель)  Александра Куликова  .  . остальные учителя  . |
| е.  учителя, отвечающие за самые дорогие экскурсии | Teacher (Учитель)  Михаил Сидоров | Teacher (Учитель)  Елена Петрова  .  . остальные учителя  . |
| ж.  Экскурсии, на которые ездили все учащиеся | excursion\_name (Название экскурсии)  Музей естественной истории | excursion\_name (Название экскурсии)  Музей истории религии  .  . остальные экскурсии  . |

**5. Разработанный код для вывода данных:**

**Учащиеся, которые не ездили в музей истории религии, но ездили в Эрмитаж**

С использованием оператора NOT IN:

SELECT s.student\_name

FROM student s

JOIN list\_of\_participants pl ON s.student\_id = pl.student\_id

JOIN excursion e ON pl.excursion\_id = e.excursion\_id

WHERE e.excursion\_name = 'Эрмитаж'

AND s.student\_id NOT IN (

SELECT s.student\_id

FROM student s

JOIN list\_of\_participants pl ON s.student\_id = pl.student\_id

JOIN excursion e ON pl.excursion\_id = e.excursion\_id

WHERE e.excursion\_name = 'Музей истории религии');

С использованием оператора EXCEPT:

SELECT s.\*

FROM student AS s

JOIN list\_of\_participants AS pl

ON s.student\_id = pl.student\_id

JOIN excursion AS e

ON pl.excursion\_id = e.excursion\_id

WHERE e.excursion\_name = 'Эрмитаж'

EXCEPT

SELECT s.\*

FROM student AS s

JOIN list\_of\_participants AS pl

ON s.student\_id = pl.student\_id

JOIN excursion AS e

ON pl.excursion\_id = e.excursion\_id

WHERE e.excursion\_name = 'Музей истории религии';

С использованием левого соединения:

SELECT s.\*

FROM student AS s

JOIN list\_of\_participants AS pl

ON s.student\_id = pl.student\_id

JOIN excursion AS e

ON pl.excursion\_id = e.excursion\_id

LEFT JOIN (

SELECT pl.student\_id

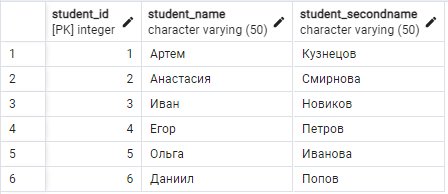
FROM list\_of\_participants AS pl

JOIN excursion AS e ON pl.excursion\_id = e.excursion\_id

WHERE e.excursion\_name = 'Музей истории религии') AS r

ON s.student\_id = r.student\_id

WHERE e.excursion\_name = 'Эрмитаж' AND r.student\_id IS NULL;



**Учитель, отвечавший за число экскурсий больше среднего**

SELECT t.teacher\_name, t.teacher\_secondname,

COUNT(DISTINCT lp.excursion\_id) AS count\_excursions

FROM teacher t

LEFT JOIN list\_of\_participants lp ON t.teacher\_id = lp.teacher\_id

GROUP BY t.teacher\_id, t.teacher\_name, t.teacher\_secondname

HAVING COUNT(DISTINCT lp.excursion\_id) > (

SELECT AVG(count\_excursions)

FROM (

SELECT t.teacher\_id,

COUNT(DISTINCT lp.excursion\_id) AS count\_excursions

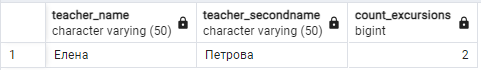
FROM teacher t

LEFT JOIN list\_of\_participants lp ON t.teacher\_id = lp.teacher\_id

GROUP BY t.teacher\_id

) AS subquery

);



**Учителя, отвечающие за самые дорогие экскурсии**

SELECT t.teacher\_name, t.teacher\_secondname

FROM teacher AS t

JOIN list\_of\_participants AS pl

ON t.teacher\_id = pl.teacher\_id

JOIN excursion AS e

ON pl.excursion\_id = e.excursion\_id AND (isavailable = 'true')

WHERE e.price = (

SELECT MAX(price) FROM excursion

);

С помощью директивы ALL:

SELECT t.teacher\_name, t.teacher\_secondname

FROM teacher AS t

JOIN list\_of\_participants AS pl

ON t.teacher\_id = pl.teacher\_id

JOIN excursion AS e

ON pl.excursion\_id = e.excursion\_id AND (isavailable = 'true')

WHERE e.price >= all (

SELECT price FROM excursion

Where price is not NULL

);



**Экскурсии, на которые ездили все учащиеся**

Реализация с NOT EXISTS:

SELECT e.excursion\_name

FROM excursion AS e

WHERE NOT EXISTS (

SELECT s.student\_id

FROM student AS s

WHERE NOT EXISTS (

SELECT \*

FROM list\_of\_participants AS pl

WHERE pl.student\_id = s.student\_id

AND pl.excursion\_id = e.excursion\_id

)

);

Реализация с COUNT:

SELECT e.excursion\_name

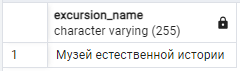
FROM excursion e

JOIN list\_of\_participants lp ON e.excursion\_id = lp.excursion\_id

JOIN student s ON lp.student\_id = s.student\_id

GROUP BY e.excursion\_id, e.excursion\_name

HAVING COUNT(DISTINCT s.student\_id) = (SELECT COUNT(\*) FROM student);



**6. Выводы:** Выполнив лабораторную работу №6 я изучил способы реализации запросов и запросов с подзапросами.