ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6  Разработка SQL запросов: запросы с подзапросами |
| по дисциплине: [Проектирование](https://pro.guap.ru/inside/students/subjects/3154777) баз данных |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Шумков Д.В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Цель работы:

1)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

Вариант 25

Багтрекинг: проекты, баги, тестировщики, разработчики (тестировщики не исправляют)

а. Баги, в названии которых содержится слово «индекс», но оно не последнее

б. Баги, найденные в одном проекте, но разными тестировщиками

в. Проект без багов

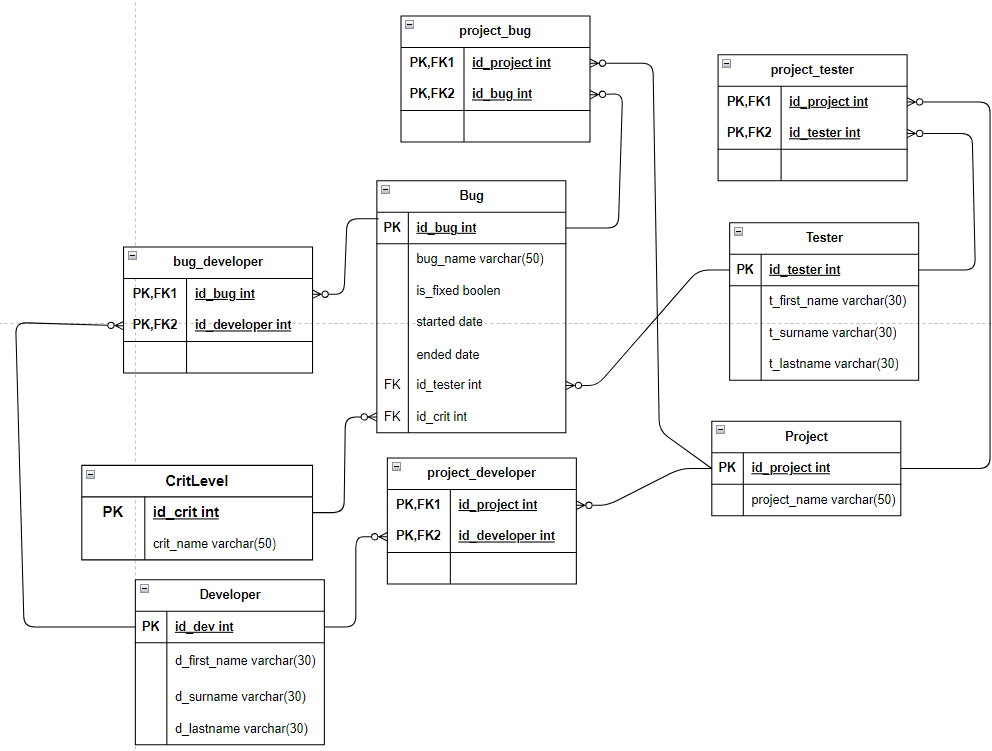
г. Критический баг, который добавили первым

д. Тестировщик, с самым большим количеством добавленных багов

е. Разработчик, у которого нет неисправленных багов

ж. Тестировщик, работавший со всеми проектами

Физическая модель БД



**Таблицы тестовых данных**

|  |  |  |
| --- | --- | --- |
| Текст запроса | Данные удовлетворяющие условиям запросов | Данные не удовлетворяющие условиям запросов |
| А. Баги, в названии которых содержится слово «индекс», но оно не последнее | Bug(bug\_name)  Индекс с неправильным типом данных,  Индекс с недействительными ссылками | Bug(bug\_name)  Все остальное |
| б. Баги, найденные в одном проекте , но разными тестировщиками | Project\_bug(id\_bug, id bug)  1,2  3,6  5,6 | Project\_bug(id\_bug, id\_bug)  Все остальное |
| в. Проект без багов | Project\_bug(id\_project)  4 | Project\_bug(id\_project)  Все остальные |
| г. Критический баг, который добавили первым | Bug(bug\_name)  Отсутствующий индекс | Bug(bug\_name)  Все остальное |
| д. Тестировщик, с самым большим количеством добавленных багов | Bug(id\_tester) tester( t\_first\_name, t\_surname, t\_lastname)  1 Андрей Петров Иванович | Bug(id\_tester)tester( t\_first\_name, t\_surname, t\_lastname)  Все кроме 1 |
| е. Разработчик, у которого нет неисправленных багов | Bug\_developer(id\_developer) developer( d\_first\_name, d\_surname, d\_lastname)  3 Алексей Петров Сергеевич | Bug\_developer(id\_developer)developer d\_first\_name, d\_surname, d\_lastname)  Все остальные |
| ж. Тестировщик, работавший со всеми проектами | Project\_tester(id\_tester, t\_first\_name, t\_surname, t\_lastname)  1 Андрей Петров Иванович | Project\_tester(id\_tester, t\_first\_name, t\_surname, t\_lastname)  Все кроме 1 |

**г. Критический баг, который добавили первым**

1) С использованием агрегатных функций

SELECT id\_bug, bug\_name

FROM bug

WHERE started = (

SELECT MIN(started)

FROM bug

WHERE id\_crit = (SELECT id\_crit FROM CritLevel WHERE crit\_name = 'Критический')

);

**д. Тестировщик, с самым большим количеством добавленных багов**

С 2 not exists

SELECT t.id\_tester, t.t\_first\_name, t.t\_surname, t.t\_lastname

FROM tester t

WHERE NOT EXISTS (

SELECT 1

FROM bug b1

WHERE b1.id\_tester = t.id\_tester

) OR NOT EXISTS (

SELECT 1

FROM bug b2

WHERE b2.id\_tester = t.id\_tester

GROUP BY b2.id\_tester

HAVING COUNT(b2.id\_bug) < (

SELECT COUNT(b3.id\_bug)

FROM bug b3

WHERE b3.id\_tester = t.id\_tester

)

)

ORDER BY (

SELECT COUNT(b.id\_bug)

FROM bug b

WHERE b.id\_tester = t.id\_tester

) DESC

LIMIT 1;

С использованием агрегатных функций

SELECT id\_tester, t\_first\_name, t\_surname, t\_lastname

FROM tester

WHERE id\_tester = (

SELECT id\_tester

FROM bug

GROUP BY id\_tester

ORDER BY COUNT(id\_bug) DESC

LIMIT 1

);

**е. Разработчик, у которого нет неисправленных багов**

**С использованием not in**

SELECT d.\* FROM developer d

WHERE d.id\_developer NOT IN (

SELECT bd.id\_developer

FROM bug\_developer bd

JOIN bug b ON bd.id\_bug = b.id\_bug

WHERE b.is\_fixed = FALSE

);

**С использованием except**

SELECT d.\* FROM developer d

EXCEPT SELECT d.\* FROM developer d

JOIN bug\_developer bd ON d.id\_developer = bd.id\_developer

JOIN bug b ON bd.id\_bug = b.id\_bug

WHERE b.is\_fixed = FALSE;

**Левое соединение**

SELECT d.\*

FROM developer d

LEFT JOIN (

SELECT bd.id\_developer

FROM bug\_developer bd

JOIN bug b ON bd.id\_bug = b.id\_bug

WHERE b.is\_fixed = FALSE

) unresolved\_bugs ON d.id\_developer = unresolved\_bugs.id\_developer

WHERE unresolved\_bugs.id\_developer IS NULL;

**ж. Тестировщик, работавший со всеми проектами**

SELECT t.\*

FROM tester t

WHERE t.id\_tester NOT IN (

SELECT pt.id\_tester

FROM project\_tester pt

GROUP BY pt.id\_tester

HAVING COUNT(DISTINCT pt.id\_project) < (

SELECT COUNT(\*) FROM project

)

);

**С except**

SELECT id\_tester, t\_first\_name, t\_surname, t\_lastname FROM tester

EXCEPT

SELECT t.id\_tester, t.t\_first\_name, t.t\_surname, t.t\_lastname

FROM tester t

JOIN project\_tester pt ON t.id\_tester = pt.id\_tester

GROUP BY t.id\_tester

HAVING COUNT(DISTINCT pt.id\_project) < (SELECT COUNT(\*) FROM project);

**С левым соединением**

SELECT t.\* FROM tester t

LEFT JOIN project\_tester pt ON t.id\_tester = pt.id\_tester

GROUP BY t.id\_tester

HAVING COUNT(DISTINCT pt.id\_project) >= (SELECT COUNT(\*) FROM project);