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 №7  Хранимые процедуры |
| 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:

создать в БД хранимые процедуры, реализующие:  
— вставку данных  
— удаление данных  
— вычисление и возврат значения агрегатной функции

— формирование статистики во временной таблице.

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

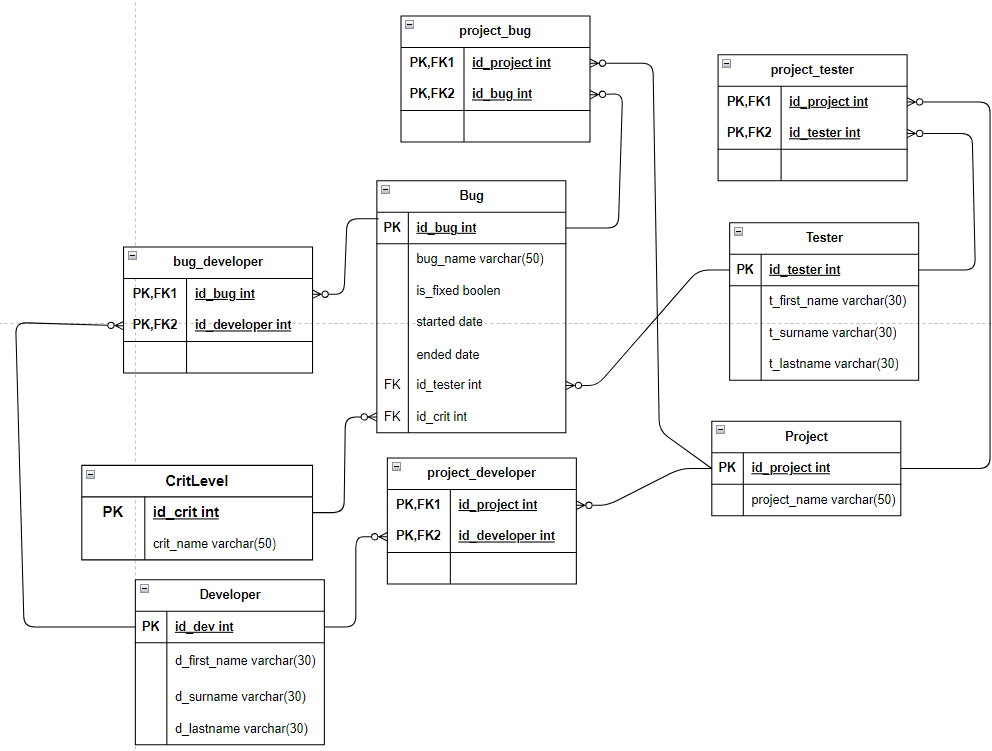
г. Критический баг, который добавили первым

д. Тестировщик, с самым большим количеством добавленных багов

е. Разработчик, у которого нет неисправленных багов

ж. Тестировщик, работавший со всеми проектами

Физическая модель БД



**Хранимая процедура с добавлением нового тестировщика.**

DELIMITER //

CREATE PROCEDURE InsertTester(

IN first\_name VARCHAR(30),

IN surname VARCHAR(30),

IN lastname VARCHAR(30)

)

BEGIN

INSERT INTO tester (t\_first\_name, t\_surname, t\_lastname)

VALUES (first\_name, surname, lastname);

END //

DELIMITER ;

**Вызов процедуры**

CALL InsertTester('Иван', 'Иванов','Иванович');

**Хранимая процедура удаления тестировщика**

DELIMITER //

CREATE PROCEDURE DeleteTester(

IN tester\_id INT

)

BEGIN

DELETE FROM project\_tester WHERE id\_tester = tester\_id;

DELETE FROM bug WHERE id\_tester = tester\_id;

DELETE FROM tester WHERE id\_tester = tester\_id;

END //

DELIMITER ;

**Вызов процедуры**

CALL DeleteTester(5);

**Хранимая процедура с возвратом значения**

DELIMITER //

CREATE PROCEDURE TotalBugsForTester(IN tester\_id INT, OUT total\_bugs INT)

BEGIN

DECLARE bug\_count INT;

SELECT COUNT(\*) INTO bug\_count

FROM bug

WHERE id\_tester = tester\_id;

SET total\_bugs = bug\_count;

END //

DELIMITER ;

**Вызов процедуры**

CALL TotalBugsForTester(1,@cnt);

SELECT @cnt AS bug\_count;



**Процедура формирования статистики во временной таблице**

DELIMITER //

CREATE PROCEDURE GenerateStatistics()

BEGIN

CREATE TEMPORARY TABLE temp\_statistics (

tester\_id INT,

total\_bugs INT

);

INSERT INTO temp\_statistics (tester\_id, total\_bugs)

SELECT id\_tester, COUNT(\*) AS total\_bugs

FROM bug

GROUP BY id\_tester;

SELECT \* FROM temp\_statistics;

DROP TEMPORARY TABLE IF EXISTS temp\_statistics;

END //

DELIMITER ;

**Вызов процедуры**

CALL GenerateStatistics();

