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:

По аналогии с примерами, приведенными в п. 1 реализовать для своей базы данныхтриггеры для всех событий (insert,delete,update) до и после. Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение из 3 предложенных. Вычислимые поля можно добавить при необходимости.

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

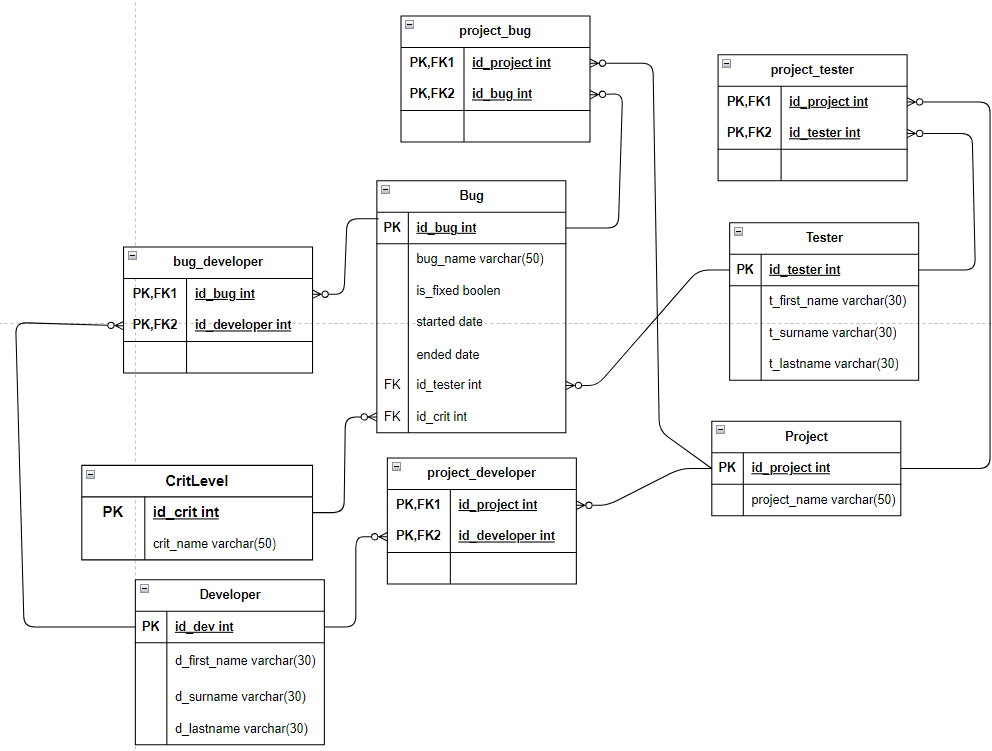
г. Критический баг, который добавили первым

д. Тестировщик, с самым большим количеством добавленных багов

е. Разработчик, у которого нет неисправленных багов

ж. Тестировщик, работавший со всеми проектами

Физическая модель БД



-- Логирование

DELIMITER //

CREATE TRIGGER log\_bug\_changes\_on\_update

AFTER UPDATE ON bug

FOR EACH ROW

BEGIN

-- Запись изменений в лог-таблицу

INSERT INTO bug\_log (bug\_id, action, timestamp)

VALUES (NEW.id\_bug, CONCAT('Action: UPDATE'), NOW());

END;

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER log\_bug\_changes

AFTER INSERT ON bug

FOR EACH ROW

BEGIN

-- Запись изменений в лог-таблицу

INSERT INTO bug\_log (bug\_id, action, timestamp)

VALUES (NEW.id\_bug, CONCAT('Action: INSERT'), NOW());

END;

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER log\_bug\_changes\_on\_delete

AFTER DELETE ON bug

FOR EACH ROW

BEGIN

-- Запись изменений в лог-таблицу

INSERT INTO bug\_log (bug\_id, action, timestamp)

VALUES (OLD.id\_bug, CONCAT('Action: DELETE'), NOW());

END;

//

DELIMITER ;

CREATE TABLE bug\_log

(

log\_id INT PRIMARY KEY AUTO\_INCREMENT,

bug\_id INT,

action VARCHAR(255),

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (bug\_id) REFERENCES bug(id\_bug) ON DELETE CASCADE ON UPDATE CASCADE

);

-- Обеспечение безопасности данных

DELIMITER //

CREATE TRIGGER ensure\_data\_security

BEFORE INSERT ON tester

FOR EACH ROW

BEGIN

-- Предотвращение вставки данных с определенными условиями

IF NEW.t\_first\_name = 'запрещенное\_имя' THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Запрещенное имя тестировщика';

END IF;

END;

//

DELIMITER ;

--Триггеры для обеспечения ссылочной целостности

DELIMITER //

CREATE TRIGGER enforce\_foreign\_key\_before\_delete

BEFORE DELETE ON tester

FOR EACH ROW

BEGIN

IF (SELECT COUNT(\*) FROM bug WHERE id\_tester = OLD.id\_tester) > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Нельзя удалить тестировщика';

END IF;

END;

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER enforce\_foreign\_key\_project\_tester

BEFORE INSERT ON project\_tester

FOR EACH ROW

BEGIN

IF NOT EXISTS (SELECT 1 FROM project WHERE id\_project = NEW.id\_project) OR

NOT EXISTS (SELECT 1 FROM tester WHERE id\_tester = NEW.id\_tester) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Не существует проекта или тестировщика';

END IF;

END;

//

DELIMITER ;