Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное

образовательное учреждение высшего образования

Санкт-Петербургский университет аэрокосмического приборостроения

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8  «Триггеры. Обеспечение активной целостности данных базы данных» |
|  |
| *по дисциплине: ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ* |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4136 |  |  |  | Сурнина А.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт – Петербург

2023

ЗАДАНИЕ: Реализовать для своей базы данных триггеры для всех событий (insert,delete, update) до и после.(6 триггеров) Часть из которых будет обеспечивать ссылочную целостность, остальные могут иметь другое назначение из других предложенных , но не менее 2 различных  
(- Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей);  
- логирование (запись) изменений;  
- обеспечения безопасности данных;  
- логическое (мягкое) удаление данных  
- проверка корректности проводимых действий.).  
Вычислимые поля можно добавить при необходимости.

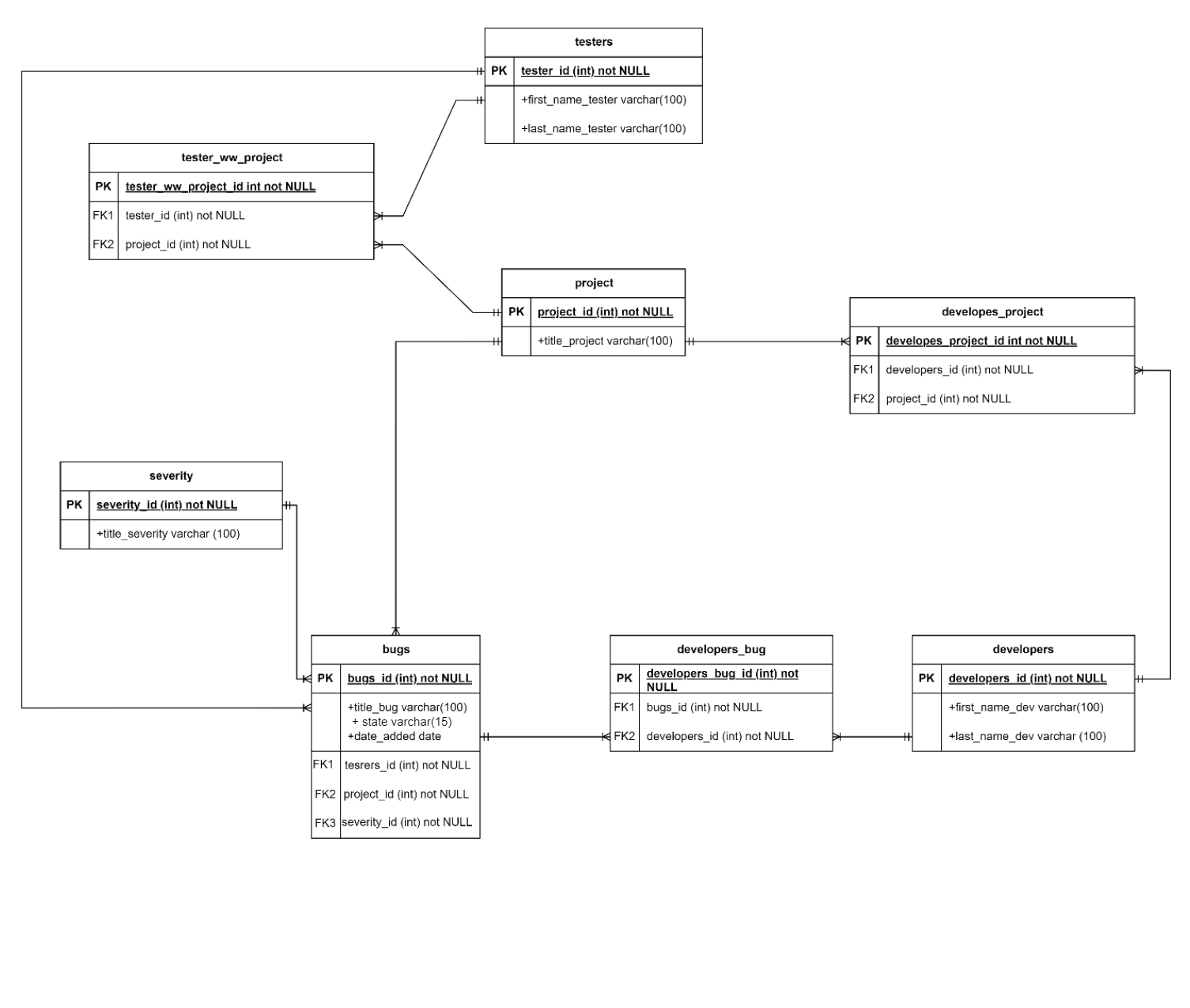
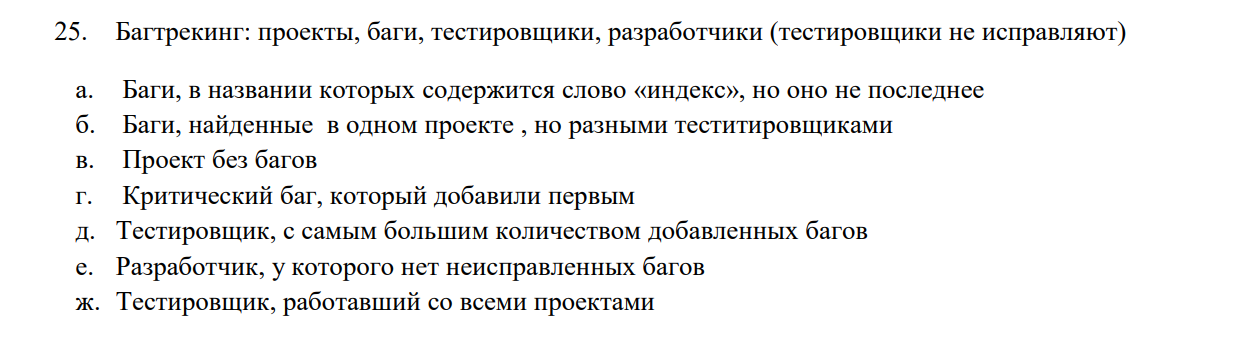
Содержание отчета:

— Текст задания.

— физическую модель БД.

— назначение и тексты триггеров;

— SQL операторы и наборы данных, иллюстрирующие работу триггеров.

Вариант:

1)DELIMITER //

CREATE TRIGGER trg\_prevent\_delete\_project\_with\_open\_bugs

BEFORE DELETE ON project

FOR EACH ROW

BEGIN

IF EXISTS (SELECT \* FROM bugs WHERE project\_id = OLD.project\_id AND state = 'не исправлен') THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Нельзя удалить проект с открытыми багами';

END IF;

END;

//

DELIMITER ;

delete from project where project\_id = 2;





Таблица bugs:

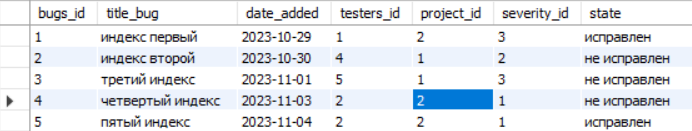
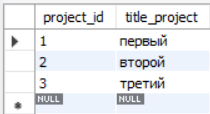
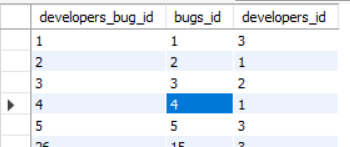


Таблица project:



Так же есть таблица developers\_bug:



2) **AFTER INSERT** Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей)

DELIMITER //

CREATE TRIGGER trg\_after\_insert\_tester\_ww\_project

AFTER INSERT ON tester\_ww\_project

FOR EACH ROW

BEGIN

-- Увеличиваем счетчик тестеров в проекте после добавления связи в tester\_ww\_project

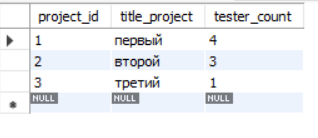
UPDATE project

SET tester\_count = tester\_count + 1

WHERE project\_id = NEW.project\_id;

END; //

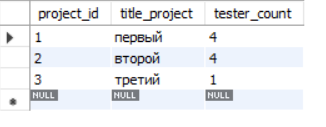
DELIMITER ;



Вставляем данные:

insert into tester\_ww\_project values (11, 3, 2);  
изменения в таблице project:

(в таблицу project добавила столбец tester\_count)



3) **Before insert -** обеспечения безопасности данных

DELIMITER //

CREATE TRIGGER trg\_before\_insert\_bug

BEFORE INSERT ON bugs

FOR EACH ROW

BEGIN

-- Проверка корректности проводимых действий: баг может быть добавлен только для существующего проекта

IF NOT EXISTS (SELECT \* FROM project WHERE project\_id = NEW.project\_id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Проект с указанным project\_id не существует';

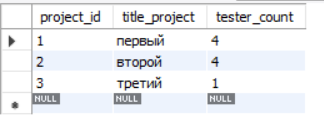
END IF;

END;

//

DELIMITER ;

Попытаемся вставить баг с project\_id = 5 (у меня всего 3)



insert into bugs values (10, 'невыерный', '2024-01-06', 8, 5, 2, 'неверный');



4) **after delete** Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей)

DELIMITER //

CREATE TRIGGER trg\_after\_delete\_ww\_project

AFTER DELETE ON tester\_ww\_project

FOR EACH ROW

BEGIN

-- Уменьшаем счетчик тестеров в проекте после удаления связи в tester\_ww\_project

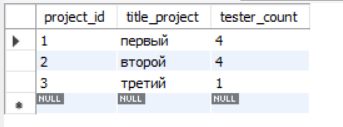
UPDATE project

SET tester\_count = tester\_count - 1

WHERE project\_id = OLD.project\_id;

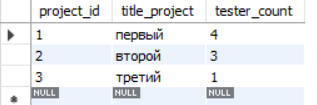
END; //

DELIMITER ;



Удаляем нашего тестировщика из таблицы tester\_ww\_project

delete from tester\_ww\_project where testers\_id =3;

Изменения в таблице project:  


5) BEFORE UPDATE - обеспечения безопасности данных

DELIMITER //

CREATE TRIGGER trg\_check\_bug\_before\_update

BEFORE UPDATE ON developers\_bug

FOR EACH ROW

BEGIN

IF NOT EXISTS (SELECT 1 FROM bugs WHERE bugs\_id = NEW.bugs\_id) THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Баг с указанным bugs\_id не существует';

END IF;

END;

//

DELIMITER ;

update developers\_bug

set bugs\_id = 100

where developers\_bug\_id = 5;





6) AFTER UPDATE - Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей)

DELIMITER //

CREATE TRIGGER trg\_after\_update\_tester\_ww\_project

AFTER UPDATE ON tester\_ww\_project

FOR EACH ROW

BEGIN

-- Обновляем tester\_count в таблице project после изменения project\_id

IF NEW.project\_id <> OLD.project\_id THEN

UPDATE project

SET tester\_count = tester\_count - 1

WHERE project\_id = OLD.project\_id;

UPDATE project

SET tester\_count = tester\_count + 1

WHERE project\_id = NEW.project\_id;

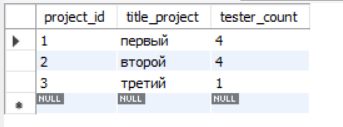
END IF;

END;

//

DELIMITER ;

Было:



Меняем project\_id в таблице tester\_ww\_project

update tester\_ww\_project

set project\_id = 3

where tester\_ww\_project\_id = 11;  
