Задание 4. Создайте хранимые процедуры

1. **Подсчёт количества студентов в группе:**

CREATE OR REPLACE FUNCTION count\_students\_in\_group(

IN p\_group\_code varchar(20)

)

RETURNS integer

AS $$

DECLARE

student\_count integer;

BEGIN

SELECT COUNT(\*) INTO student\_count

FROM studying\_student

WHERE group\_code = p\_group\_code;

RETURN student\_count;

END;

$$ LANGUAGE plpgsql;

SELECT count\_students\_in\_group('А11-0');

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

1. **Подсчёт количества студентов, успешно сдавших экзамен по дисциплине (на 4 и 5):**

CREATE OR REPLACE FUNCTION count\_students\_passed\_exam(

IN p\_discipline\_name varchar(60)

)

RETURNS integer

AS $$

DECLARE

student\_count integer;

BEGIN

SELECT COUNT(\*) INTO student\_count

FROM exam e

JOIN discipline\_on\_SP ds ON e.exam\_id = ds.exam\_id

WHERE ds.discipline\_name = p\_discipline\_name

AND e.mark >= 4;

RETURN student\_count;

END;

$$ LANGUAGE plpgsql;

SELECT count\_students\_passed\_exam('программирование');

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

1. **Расписание экзаменов для дисциплины:**

CREATE OR REPLACE FUNCTION get\_exam\_schedule\_for\_discipline(

IN p\_discipline\_name varchar(60) )

RETURNS TABLE(exam\_date date, teacher\_name character varying(50), classroom\_number integer)

AS $$

BEGIN

RETURN QUERY

SELECT es.exam\_date, (t.lastname || ' ' || t.teachers\_name)::character varying(50) AS teacher\_name,

c.classroom\_number

FROM exam\_schedule es

JOIN exam e ON es.exam\_id = e.exam\_id

JOIN teacher t ON e.teacher\_code = t.teacher\_code

JOIN classroom c ON es.classroom\_code = c.classroom\_code

JOIN discipline\_on\_SP ds ON e.exam\_id = ds.exam\_id

WHERE ds.discipline\_name = p\_discipline\_name

ORDER BY es.exam\_date;

END;

$$ LANGUAGE plpgsql;

SELECT \* FROM get\_exam\_schedule\_for\_discipline('программирование');Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Задание 5. Триггеры

**Триггер для проверки записи оценки в экзамене:**

CREATE OR REPLACE FUNCTION check\_exam\_mark()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.mark < 2 OR NEW.mark > 5 THEN

RAISE EXCEPTION 'Оценка должна быть в диапазоне от 2 до 5';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER exam\_mark\_check

BEFORE INSERT ON exam

FOR EACH ROW

EXECUTE FUNCTION check\_exam\_mark();

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Автоматически созданное описание

Вводим ошибочный инсерт:

INSERT INTO exam (mark, exam\_code, exam\_id, attestation\_type, try\_number, teacher\_code, Sstudent\_code)

VALUES (1, 'EXAM123', 'EXAM001', 'Экзамен', 'Первая попытка', 22, 'STUD001');

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание