

федеральное государственное автономное образовательное учреждение высшего  
образования  
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

**Лабораторная работа №3**  
**«Процедуры, функции, триггеры в**  
**PostgreSQL» по дисциплине**  
**«Проектирование и**  
**реализация баз данных»**

**Выполнил:**  
студент 2 курса ФИКТ  
группы K3241  
Тимофеев Н.А.

**Проверил:**  
Говорова М.В.

Санкт-Петербург  
2022

**Цель работы:** овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, pgadmin 4.

**Практическое задание:**

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
2. Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5). Допустимо создать универсальный триггер или отдельные триггеры на логирование действий.

**Индивидуальное задание. Вариант 5. БД «Издательство компьютерной литературы»**

Описание предметной области: Издательство занимается выпуском литературы по различным областям информатики. Покупатели книг приобретают книги на базе издательства. Когда на базе заканчиваются книги, издается дополнительный тираж. В каждом заказе заказчик может заказать разную литературу.

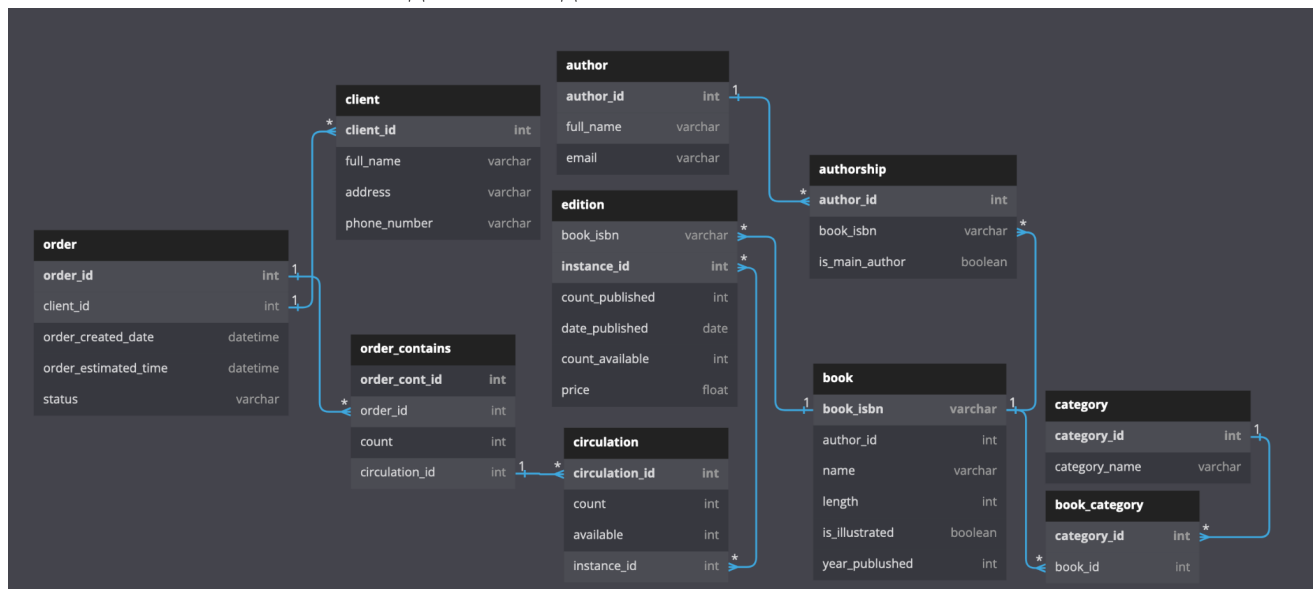
БД должна содержать следующий минимальный набор сведений: Фамилия автора. Имя автора. Отчество автора. Код автора. E-mail автора. Код ISBN. Название книги. Количество страниц. Наличие иллюстраций. Код категории книги. Категория книги. Количество страниц. Год начала издания. Розничная цена книги. Тираж. Дата тиража. Количество экземпляров на базе издательства. Код заказчика. Фамилия заказчика. Имя заказчика. Отчество заказчика. Адрес заказчика. Телефон заказчика. Код заказа. Дата заказа. Срок заказа. Количество экземпляров книги в заказе. Статус заказа.

**1. БД «Издательство компьютерной литературы»**

**2. Сущности:**

- a. Писатель (Код автора (PK), ФИО автора, Электронная почта автора)
- b. Книга (Код ISBN (PK), название книги, количество страниц, наличие иллюстраций, год начала издания)
- c. Категория (Код категории книги (PK), название категории)
- d. Издание (Номер издания (PK), дата тиража, Код ISBN, тираж, кол-во экземпляров на базе издательства, код заказа, розничная цена книги)
- e. Заказчик (Код заказчика (PK), ФИО заказчика, адрес заказчика, телефон заказчика)
- f. Заказ (Код заказа (PK), код заказчика, дата заказа, срок заказа, статус заказа)
- g. Состав заказа (Номер состава заказа (PK), номер заказа, номер издания, кол-во экземпляров книги в заказе)

### 3. Схема логической модели базы данных



### 4. Хранимые процедуры

- Для снижения цен на книги, которые находятся на базе в количестве, превышающем 1000 штук.

```
CREATE OR REPLACE PROCEDURE make_discount() LANGUAGE plpgsql as $$
BEGIN
    UPDATE edition e
    SET price = price * 0.90
    WHERE instance_id IN (SELECT instance_id FROM circulation WHERE available > 100);
    COMMIT;
END;
$$;

call make_discount();
```

Цены до обновления:

	book_isbn	instance_id	date_published	price
1	1	1	2002-10-02	1000
2	1	2	2023-10-02	1400
3	2	3	2019-11-02	990
4	2	4	2022-03-02	9990
5	4	7	2019-10-09	2900
6	3	6	2017-10-02	1600
7	3	5	2015-10-02	1000

Цены после обновления:

	book_isbn	instance_id	date_published	price
1	1	1	2002-10-02	1000
2	1	2	2023-10-02	1400
3	2	3	2019-11-02	990
4	2	4	2022-03-02	9990
5	4	7	2019-10-09	2900
6	3	5	2015-10-02	900
7	3	6	2017-10-02	1440

- Для ввода новой книги.

```
CREATE OR REPLACE PROCEDURE insert_book(name varchar, length int, illustrated bool, date_pub int)
LANGUAGE plpgsql AS $$
BEGIN
    INSERT INTO book
    VALUES ((select max(book_isbn)::int + 1 FROM book), name, length, illustrated, date_pub);
END;
$$;

call insert_book('Java SE For Beginners', 1090, true, 2019);
```

	book_isbn	name	length	is_illustrated	year_published
1	1	Python programming	1120	false	2022
2	2	Art and science	122	true	2021
3	3	Popular Mechanics	320	true	2020
4	4	Guide on ORM in Python	220	true	2022
5	5	PostgreSQL v13 Handbook	1200	true	2019
6	6	Clean Architecture	670	true	2022
7	7	Java SE For Beginners	1090	true	2019

- Для ввода нового заказа.

```
CREATE OR REPLACE PROCEDURE new_order(client_name varchar, estimated_deliver_date interval = '10 days'::interval)
LANGUAGE plpgsql AS $$
BEGIN
    INSERT INTO "order"
    VALUES ((SELECT max(order_id) + 1 FROM "order"),
            (SELECT c.client_id FROM client c WHERE full_name = client_name),
            CURRENT_DATE::date, CURRENT_DATE + estimated_deliver_date, 'process');
END;
$$;

call new_order('Ivan Semenov', '7 days');
```

7	7	1	2022-09-21	2022-09-29	shipping
8	8	2	2022-02-21	2022-02-28	finished
9	9	4	2022-10-02	2022-10-12	process
10	10	1	2022-10-04	2022-10-11	process

## 5. Триггеры

- Создадим триггер для логгирования изменений в таблице book

```

CREATE OR REPLACE FUNCTION logging() RETURNS TRIGGER AS
$$
DECLARE
    log_base varchar(30);
    book_name character varying;
    book_year character varying;
    log_str varchar(254);
BEGIN
    IF TG_OP = 'INSERT' THEN
        book_name = NEW.name;
        book_year = NEW.year_published;
        log_base := 'New book added';
        log_str := concat(log_base, ': ', book_name, '(', book_year, ')');
        INSERT INTO logs(text, time)
        values (log_str, CURRENT_DATE);
        RETURN NEW;
    ELSIF TG_OP = 'UPDATE' THEN
        book_name = OLD.name;
        book_year = OLD.year_published;
        log_base := 'Book updated';
        log_str := concat(log_base, ': ', book_name, '(', book_year, ')');
        INSERT INTO logs(text, time)
        values (log_str, CURRENT_DATE);
        RETURN NEW;
    ELSIF TG_OP = 'DELETE' THEN
        book_name = OLD.name;
        book_year = OLD.year_published;
        log_base := 'Book removed';
        log_str := concat(log_base, ': ', book_name, '(', book_year, ')');
        INSERT INTO logs(text, time)
        values (log_str, CURRENT_DATE);
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE plpgsql;





```

```

CREATE TRIGGER book_logging
    AFTER INSERT OR UPDATE OR DELETE
    ON
        book
    FOR EACH ROW
EXECUTE PROCEDURE logging();

```

### Результат работы триггера:

	 id  1	 text	 time
1	1	New book added: New book(1111)	2022-10-04
2	2	New book added: Byte of python(2017)	2022-10-04
3	3	New book added: Grokking algorithms(2016)	2022-10-04
4	4	Book removed: Byte of python(2017)	2022-10-04
5	5	Book updated: New book(1111)	2022-10-04

**Вывод:** Я научился создавать, модифицировать и использовать хранимые процедуры и триггеры (на примере PostgreSQL)