

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»

Выполнила:
студентка 2 курса ФИКТ
группы K3241
Барышева З.А.

Проверила:
Говорова М.В.

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.

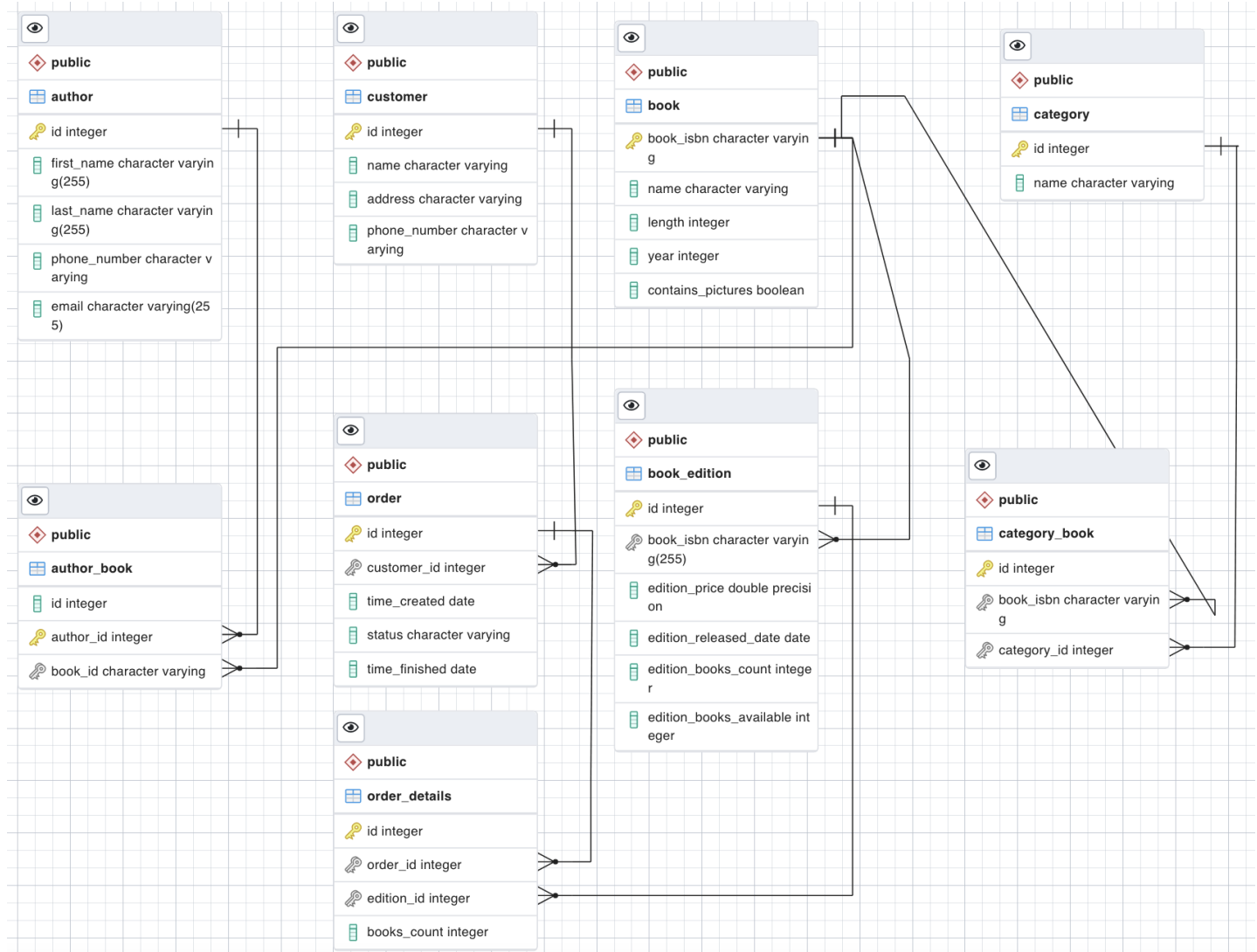
БД должна содержать следующий минимальный набор сведений: Фамилия автора. Имя автора. Отчество автора. Код автора. E-mail автора. Код ISBN. Название книги. Количество страниц. Наличие иллюстраций. Код категории книги. Категория книги. Количество страниц. Год начала издания. Розничная цена книги. Тираж. 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:

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

3. ERD-диаграмма



4. Хранимые процедуры:

- Для снижения цен на книги, которые находятся на базе в количестве, превышающем 100 штук.

Query	Query History
1	CREATE OR REPLACE PROCEDURE make_discount()
2	LANGUAGE plpgsql as
3	\$\$
4 ▼	BEGIN
5	UPDATE book_edition e
6	SET edition_price = edition_price - 150
7	WHERE e.edition_books_available > 100;
8	COMMIT ;
9	END ;
10	\$\$;

До:

11 **SELECT** * **FROM** book_edition;

Data output Messages Explain × Notifications

	id [PK] integer	book_isbn character varying (255)	edition_price double precision	edition_released_date date	edition_books_count integer	edition_books_available integer
1	1	1	1300	2001-10-09	3500	3000
2	2	1	2999	2012-10-09	1200	18
3	3	2	120	2019-10-09	100	100
4	4	3	1999	2016-10-09	1000	987
5	5	4	5800	2002-10-09	1500	1211

После:

12 **SELECT** * **FROM** book_edition;

Data output Messages Explain × Notifications

	id [PK] integer	book_isbn character varying (255)	edition_price double precision	edition_released_date date	edition_books_count integer	edition_books_available integer
1	2	1	2999	2012-10-09	1200	18
2	3	2	120	2019-10-09	100	100
3	1	1	1150	2001-10-09	3500	3000
4	4	3	1849	2016-10-09	1000	987
5	5	4	5650	2002-10-09	1500	1211

- Для ввода новой книги.

Query Query History

```
1 DROP PROCEDURE insert_book(character varying,integer,integer);
2 CREATE OR REPLACE PROCEDURE insert_book(name varchar, length int, year_created int)
3     LANGUAGE plpgsql AS
4 $$
5 BEGIN
6     INSERT INTO book
7     VALUES ((select max(book_isbn)::int + 1 FROM book), name, length, year_created, false);
8 END;
9 $$;
```

До:

12 **SELECT** * **FROM** book;

Data output Messages Explain × Notifications

	book_isbn [PK] character varying	name character varying	length integer	year integer	contains_pictures boolean
1	1	1984	585	1949	true
2	2	Мысли о России	10	2016	false
3	3	A tour of C++	890	2013	true
4	5	Преступление и ...	500	1866	false
5	4	Ради удовольст...	350	2022	true
6	6	Три мушкетера	440	1923	false

После:

```
11 call insert_book('Коты воители', 1000, 2022);
12 SELECT * FROM book;
```

Data output Messages Explain X Notifications

	book_isbn [PK] character varying	name character varying	length integer	year integer	contains_pictures boolean
1	1	1984	585	1949	true
2	2	Мысли о России	10	2016	false
3	3	A tour of C++	890	2013	true
4	5	Преступление и наказан...	500	1866	false
5	4	Ради удовольствия	350	2022	true
6	6	Три мушкетера	440	1923	false
7	7	Коты воители	1000	2022	false

- Для ввода нового заказа.

Query Query History

```
1 CREATE OR REPLACE PROCEDURE new_order(client_name varchar, estimated_deliver_date date)
2 LANGUAGE plpgsql AS
3 $$
4 BEGIN
5     INSERT INTO "order"
6     VALUES ((SELECT max(id) + 1 FROM "order"),
7             (SELECT c.id FROM customer c WHERE name = client_name),
8             CURRENT_DATE, 'оплачено', estimated_deliver_date);
9 END;
10 $$;
11
12 call new_order('Alex', (CURRENT_DATE + INTERVAL '10 days')::date);
13
```

До:

```
13 SELECT * FROM "order";
```

Data output Messages Explain X Notifications

	id [PK] integer	customer_id integer	time_created date	status character varying	time_finished date
1	1	1	2022-09-09	доставлено	2022-09-21
2	2	1	2022-09-27	доставлено	2022-10-04
3	3	2	2022-09-04	доставлено	2022-09-17
4	4	2	2022-10-01	доставлено	2022-10-08
5	5	1	2022-10-06	отправлено	[null]
6	6	1	2022-10-04	доставлено	2022-10-08

После:

	id [PK] integer	customer_id integer	time_created date	status character varying	time_finished date
1	1	1	2022-09-09	доставлено	2022-09-21
2	2	1	2022-09-27	доставлено	2022-10-04
3	3	2	2022-09-04	доставлено	2022-09-17
4	4	2	2022-10-01	доставлено	2022-10-08
5	5	1	2022-10-06	отправлено	[null]
6	6	1	2022-10-04	доставлено	2022-10-08
7	7	1	2022-10-09	оплачено	2022-10-19

5. Триггер:

Создадим таблицу для логгирования действий с заказами:

Query	Query History
1	CREATE TABLE "logs" (
2	"id" serial PRIMARY KEY,
3	"method" varchar,
4	"text" varchar,
5	"time" date
6);
7	

Определим процедуру для триггера:

```

1 CREATE OR REPLACE FUNCTION record_orders_logs() RETURNS TRIGGER AS
2 $$
3 BEGIN
4     IF TG_OP = 'INSERT' THEN
5         INSERT INTO logs(method, text, time)
6         values (TG_OP, concat('New order: ', NEW.id,
7                                '(estimated time: ',
8                                NEW.time_finished, ')'),
9                CURRENT_DATE);
10        RETURN NEW;
11    ELSIF TG_OP = 'UPDATE' THEN
12        INSERT INTO logs(method, text, time)
13        values (TG_OP, concat('Order updated: ', NEW.id,
14                               '(status: ', NEW.status, ')'), CURRENT_DATE);
15        RETURN NEW;
16    ELSIF TG_OP = 'DELETE' THEN
17        INSERT INTO logs(method, text, time)
18        values (TG_OP, concat('Order removed: ', OLD.id,
19                               '(status: ', OLD.status,
20                               'created: ', OLD.time_created, ')'),
21                CURRENT_DATE);
22        RETURN OLD;
23    END IF;
24 END;
25 $$ LANGUAGE plpgsql;

```

Создадим триггер:

```

CREATE TRIGGER book_logging
AFTER INSERT OR UPDATE OR DELETE
ON
    "order"
FOR EACH ROW
EXECUTE PROCEDURE record_orders_logs();

```

Результат работы:

```

35 insert into "order" (id, customer_id, time_created, status, time_finished)
36 values (8, 1, CURRENT_DATE, 'оплачено', CURRENT_DATE + INTERVAL '15 days');

```

```
DELETE FROM "order" WHERE id=8;
```

4

```
5 UPDATE "order" SET status='доставлен' WHERE id=7;
```

Таблица logs:

Query

Query History

1 SELECT * FROM logs ORDER BY time;

Data output

Messages

Explain

×

Notifications

≡+

	id [PK] integer	method character varying	text character varying	time date
1	1	INSERT	New order: 8(estimated time: 2022-10-24)	2022-10-09
2	2	DELETE	Order removed: 8(status: оплачено, created: 2022-10-09)	2022-10-09
3	3	UPDATE	Order updated: 7(status: доставлен)	2022-10-09

6. **Вывод:** В ходе работы научилась использовать и модифицировать хранимые процедуры и триггеры