

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра прикладной математики

Отчет по лабораторной работе № 6

«Представления. Функции. Триггеры.»

по курсу «Базы данных»

Студент

Группа ПМ-18

Руководитель

подпись, дата

Полухина Е.Д.

фамилия, инициалы

подпись, дата

Хабибуллина Е.Л.

фамилия, инициалы

Липецк 2020 г.

Содержание

Задание кафедры	3
Примеры запросов.....	4
3 представления (1 обязательно с вычисляемыми полями).....	4
Запросы к каждому представлению (из них по 1 запросу на INSERT/ UPDATE/DELETE)	5
Функции SQL с базовыми типами (2 шт: 1 без аргументов, 1 с аргументами базовых типов)	13
Функции SQL со сложными типами	15
Функции SQL с выходными параметрами.....	16
Функции SQL с переменным числом аргументов.....	16
Функции SQL со значениями аргументов по умолчанию	17
Функции SQL, возвращающие таблицы (TABLE)	19
Функция PL/pgSQL	20
Два триггера для INSERT / UPDATE / DELETE (разные комбинации с BEFORE, AFTER и INSTEAD OF).....	21
Выводы	28

Задание кафедры

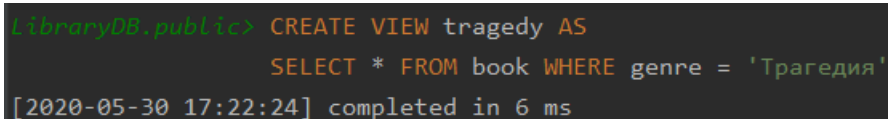
1. Написать 3 представления (1 обязательно с вычисляемыми полями)
2. Написать по 5 запросов к каждому представлению (из них по 1 запросу на INSERT/ UPDATE/DELETE)
3. Написать функции SQL с базовыми типами (2 шт: 1 без аргументов, 1 с аргументами базовых типов), привести результаты выполнения со скриншотами.
4. Написать функции SQL со сложными типами (1 шт), привести результаты выполнения со скриншотами.
5. Написать функцию SQL с выходными параметрами (использование ключевого слова OUT 1 шт.)
6. Написать функцию SQL с переменным числом аргументов (1 шт).
7. Написать функцию SQL со значениями аргументов по умолчанию (1 шт).
8. Написать функцию SQL, возвращающие таблицы (TABLE) (1 шт).
9. Создать одну функцию PL/pgSQL (обязательно наличие операторов цикла и условия)
10. Написать по два триггера для INSERT / UPDATE / DELETE (использовать разные комбинации с BEFORE, AFTER и INSTEAD OF)

Примеры запросов

3 представления (1 обязательно с вычисляемыми полями)

1. Создать представление, содержащее ISBN книги, ее название, жанр, год публикации, стоимость и количество страниц, если это трагедия.

```
CREATE VIEW tragedy AS
SELECT * FROM book WHERE genre = 'Трагедия';
```

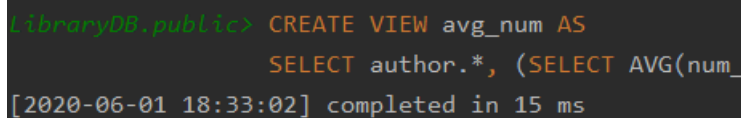


```
LibraryDB.public> CREATE VIEW tragedy AS
SELECT * FROM book WHERE genre = 'Трагедия'
[2020-05-30 17:22:24] completed in 6 ms
```

Рисунок 1 – результат запроса

2. Создать представление, содержащее id автора, ФИО автора, год его рождения и среднее количество страниц в его книгах, если его ФИО начинается с буквы «С».

```
CREATE VIEW avg_num AS
SELECT author.*, (SELECT AVG(num_pages) FROM book, book_author
WHERE book_author.id_author = book.isbn AND book_author.id_author
= author.id_author) FROM author WHERE full_name_author LIKE 'С%';
```



```
LibraryDB.public> CREATE VIEW avg_num AS
SELECT author.*, (SELECT AVG(num_
[2020-06-01 18:33:02] completed in 15 ms
```

Рисунок 2 – результат запроса

3. Создать представление, содержащее список сотрудников, чей стаж работы начинается с 2011.

```
CREATE VIEW emp_twenty_ten AS
SELECT * FROM employee WHERE seniority_begin > 2010;
```

```
libraryDB.public> CREATE VIEW emp_twenty_ten AS
SELECT * FROM employee WHERE seniority_begin > 2010
[2020-05-30 17:49:43] completed in 327 ms
```

Рисунок 3 – результат запроса

Запросы к каждому представлению (из них по 1 запросу на INSERT/ UPDATE/DELETE)

1. Вывести ФИО автора, название его книги и жанр, если это трагедия.

```
SELECT full_name_author, title, genre FROM tragedy, book_author,
author WHERE tragedy.isbn = book_author.id_book AND
author.id_author = book_author.id_author;
```

	full_name_author	title	genre
1	Щербаков Федор Михайлович	Хороший доктор	Трагедия
2	Пришвина Анна Михайловна	Сияние	Трагедия
3	Тургенев Федор Александрович	Три билета	Трагедия
4	Тургенева Ольга Васильевна	Объятия сна	Трагедия
5	Есенин Федор Александрович	Замок из стекла	Трагедия
6	Сальникова Елена Васильевна	Сладкая жизнь	Трагедия
7	Носов Сергей Сергеевич	Последний лист	Трагедия

Рисунок 4 – результат запроса

2. Вывести дату выдачи, дату сдачи, название книги и ее стоимость, если это трагедия.

```
SELECT date_in, date_return, title, cost FROM tragedy, delivery
WHERE tragedy.isbn = delivery.isbn;
```

	date_in	date_return	title	cost
1	2019-04-18	2019-05-12	Три билета	489.69
2	2004-08-18	2005-06-22	Три билета	489.69
3	2012-03-24	2012-04-17	Сияние	710.73
4	2014-05-28	2014-06-18	Последний лист	584.51
5	2005-02-18	2005-03-08	Три билета	489.69
6	2011-06-23	2011-10-18	Объятия сна	930.26
7	2002-03-13	2002-04-15	Объятия сна	930.26
8	2019-08-14	<null>	Три билета	489.69
9	2018-10-30	2018-11-17	Замок из стекла	783.86
10	2015-04-12	2015-04-27	Замок из стекла	783.86
11	1995-09-06	1995-09-23	Объятия сна	930.26

Рисунок 5 – результат запроса

3. Добавить книгу с названием «Луна» 2019 года издания, в которой 195 страниц, жанр «трагедия» и стоимость 450 и количество книг 5.

```
INSERT INTO tragedy
VALUES (31, 'Луна', 'Трагедия', 195, 2019, 450, 5);
```

	isbn	title	genre	num_pages	year_publish	cost	num_books
1	5	Хороший доктор	Трагедия	834	2015	811.72	2
2	8	Замок из стекла	Трагедия	318	1986	783.86	4
3	10	Сияние	Трагедия	250	2000	710.73	8
4	12	Сладкая жизнь	Трагедия	101	2010	667.87	4
5	14	Последний лист	Трагедия	598	1995	584.51	2
6	24	Объятия сна	Трагедия	455	2006	930.26	0
7	30	Три билета	Трагедия	378	1992	489.69	8

Рисунок 6 – До изменения

	isbn	title	genre	num_pages	year_publish	cost	num_books
1	5	Хороший доктор	Трагедия	834	2015	811.72	2
2	8	Замок из стекла	Трагедия	318	1986	783.86	4
3	10	Сияние	Трагедия	250	2000	710.73	8
4	12	Сладкая жизнь	Трагедия	101	2010	667.87	4
5	14	Последний лист	Трагедия	598	1995	584.51	2
6	24	Объятия сна	Трагедия	455	2006	930.26	0
7	30	Три билета	Трагедия	378	1992	489.69	8
8	31	Луна	Трагедия	195	2019	450.00	5

Рисунок 7 – После изменения

- Добавить 3 книги к количеству книг с названием «Объятия сна» и жанром трагедия и вывести полученный результат.

```
UPDATE tragedy SET num_books = num_books + 3 WHERE title = 'Объятия сна';
```

	isbn	title	genre	num_pages	year_publish	cost	num_books
1	5	Хороший доктор	Трагедия	834	2015	811.72	2
2	8	Замок из стекла	Трагедия	318	1986	783.86	4
3	10	Сияние	Трагедия	250	2000	710.73	8
4	12	Сладкая жизнь	Трагедия	101	2010	667.87	4
5	14	Последний лист	Трагедия	598	1995	584.51	2
6	24	Объятия сна	Трагедия	455	2006	930.26	0
7	30	Три билета	Трагедия	378	1992	489.69	8
8	31	Луна	Трагедия	195	2019	450.00	5

Рисунок 8 – До изменения

	isbn	title	genre	num_pages	year_publish	cost	num_books
1	5	Хороший доктор	Трагедия	834	2015	811.72	2
2	8	Замок из стекла	Трагедия	318	1986	783.86	4
3	10	Сияние	Трагедия	250	2000	710.73	8
4	12	Сладкая жизнь	Трагедия	101	2010	667.87	4
5	14	Последний лист	Трагедия	598	1995	584.51	2
6	30	Три билета	Трагедия	378	1992	489.69	8
7	31	Луна	Трагедия	195	2019	450.00	5
8	24	Объятия сна	Трагедия	455	2006	930.26	3

Рисунок 9 – После изменения

- Удалить книгу под названием «Луна» и с жанром трагедия.

```
DELETE FROM tragedy WHERE title = 'Луна';
```

	isbn	title	genre	num_pages	year_publish	cost	num_books
1	5	Хороший доктор	Трагедия	834	2015	811.72	2
2	8	Замок из стекла	Трагедия	318	1986	783.86	4
3	10	Сияние	Трагедия	250	2000	710.73	8
4	12	Сладкая жизнь	Трагедия	101	2010	667.87	4
5	14	Последний лист	Трагедия	598	1995	584.51	2
6	24	Объятия сна	Трагедия	455	2006	930.26	0
7	30	Три билета	Трагедия	378	1992	489.69	8
8	31	Луна	Трагедия	195	2019	450.00	5

Рисунок 10 – До изменения

	isbn	title	genre	num_pages	year_publish	cost	num_books
1	5	Хороший доктор	Трагедия	834	2015	811.72	2
2	8	Замок из стекла	Трагедия	318	1986	783.86	4
3	10	Сияние	Трагедия	250	2000	710.73	8
4	12	Сладкая жизнь	Трагедия	101	2010	667.87	4
5	14	Последний лист	Трагедия	598	1995	584.51	2
6	30	Три билета	Трагедия	378	1992	489.69	8
7	24	Объятия сна	Трагедия	455	2006	930.26	3

Рисунок 11 – После изменения

6. Вывести ФИО автора, название и жанр его книги, если его фамилия начинается с буквы «С».

```
select full_name_author, title, genre from avg_num, book_author,
book WHERE avg_num.id_author = book_author.id_author AND
book.isbn = book_author.id_book;
```

	full_name_author	title	genre
1	Сальникова Елена Васильевна	Сладкая жизнь	Трагедия
2	Сальников Федор Михайлович	Любовь и другие лекарства	Роман
3	Сальникова Ольга Михайловна	Счастье и любовь	Поэма
4	Сальникова Ольга Михайловна	Цветок жизни	Детектив

Рисунок 12 – результат запроса

7. Вывести ФИО читателя, взявшего книгу автора, чья фамилия начинается на букву «С», дату выдачи, название книги, которую он брал и фамилию автора, если книга была выдана не позднее 2010 года.


```
SELECT full_name_reader, date_in, title, full_name_author FROM
avg_num, book_author, book, delivery, reader WHERE
avg_num.id_author = book_author.id_author AND book.isbn =
book_author.id_book AND reader.id_reader = delivery.id_reader AND
delivery.isbn = book.isbn AND date_in <= '2010-12-31';
```

	full_name_reader	date_in	title	full_name_author
1	Матвеева Мария Сергеевна	2004-03-21	Счастье и любовь	Сальникова Ольга Михайловна
2	Суворов Александр Михайлович	2005-04-17	Цветок жизни	Сальникова Ольга Михайловна

Рисунок 13 – результат запроса до изменения

8. Добавить автора Симонова Романа Олеговича 1980 года рождения.

```
INSERT INTO avg_num
VALUES (31, 1980, 'Симонов Роман Олегович');
```

	id_author	year_birth	full_name_author	avg
1	17	1759	Сальникова Елена Васильевна	682
2	23	1833	Сальникова Ольга Михайловна	899
3	4	1866	Сальников Федор Михайлович	1002

Рисунок 14 – До изменения

	id_author	year_birth	full_name_author	avg
1	17	1759	Сальникова Елена Васильевна	682
2	23	1833	Сальникова Ольга Михайловна	899
3	4	1866	Сальников Федор Михайлович	1002
4	31	1980	Симонов Роман Олегович	<null>

Рисунок 15 – После изменения

9. Изменить год рождения автора с id 31 на 1978.

```
UPDATE avg_num SET year_birth = 1978 WHERE id_author = 31;
```

	id_author	year_birth	full_name_author	avg
1	17	1759	Сальникова Елена Васильевна	682
2	23	1833	Сальникова Ольга Михайловна	899
3	4	1866	Сальников Федор Михайлович	1002
4	31	1980	Симонов Роман Олегович	<null>

Рисунок 16 – До изменения

	id_author	year_birth	full_name_author	avg
1	17	1759	Сальникова Елена Васильевна	682
2	23	1833	Сальникова Ольга Михайловна	899
3	4	1866	Сальников Федор Михайлович	1002
4	31	1978	Симонов Роман Олегович	<null>

Рисунок 17 – После изменения

10. Удалить автора Симонова Романа Олеговича с id 31.

```
DELETE FROM avg_num WHERE id_author = 31;
```

	id_author	year_birth	full_name_author	avg
1	17	1759	Сальникова Елена Васильевна	682
2	23	1833	Сальникова Ольга Михайловна	899
3	4	1866	Сальников Федор Михайлович	1002
4	31	1978	Симонов Роман Олегович	<null>

Рисунок 18 – До изменения

	id_author	year_birth	full_name_author	avg
1	17	1759	Сальникова Елена Васильевна	682
2	23	1833	Сальникова Ольга Михайловна	899
3	4	1866	Сальников Федор Михайлович	1002

Рисунок 19 – После изменения

11. Вывести ФИО сотрудника и название его должности, год начала его стажа, если стаж начинается с 2011 года или выше.

```
SELECT full_name_employee, title_position, seniority_begin FROM
emp_twenty_ten, position WHERE emp_twenty_ten.id_position =
position.id_position;
```

	full_name_employee	title_position	seniority_begin
1	Цветков Александр Александрович	Охранник	2016
2	Мишин Алексей Викторович	Уборщик	2011
3	Дубкова Людмила Андреевна	Системный администратор	2015
4	Цветков Алексей Сергеевич	Библиотекарь	2015
5	Яковлева Людмила Андреевна	Бухгалтер	2020
6	Доронина Анастасия Александровна	Системный администратор	2011

Рисунок 20 – результат запроса

12. Вывести ФИО сотрудника и все его выдачи, если его стаж начинается с 2011 года или выше.

```
SELECT full_name_employee, date_in, date_return, set_date_return
FROM emp_twenty_ten, delivery WHERE emp_twenty_ten.id_employee =
delivery.id_employee;
```

	full_name_employee	date_in	date_return	set_date_return
1	Цветков Алексей Сергеевич	2005-04-17	2005-05-06	2005-05-17
2	Цветков Алексей Сергеевич	2018-01-26	2018-02-19	2018-02-28
3	Цветков Алексей Сергеевич	2004-03-21	2004-09-07	2004-05-16

Рисунок 21 – результат запроса

13. Добавить сотрудника Носова Георгия Аркадьевича с годом рождения 1995-05-12 на должность системного администратора.

```
INSERT INTO emp_twenty_ten
VALUES (31, 'Носов Георгий Аркадьевич', 2020, false, '1995-05-12',
7);
```

	id_employee	full_name_employee	seniority_begin	gender_employee	birth_day_employee	id_position
1	8	Цветков Александр Александрович	2016	<input type="checkbox"/>	1957-12-06	6
2	9	Мишин Алексей Викторович	2011	<input type="checkbox"/>	1973-09-19	3
3	11	Дубкова Людмила Андреевна	2015	<input checked="" type="checkbox"/>	1966-12-01	7
4	25	Цветков Алексей Сергеевич	2015	<input type="checkbox"/>	1995-09-25	5
5	26	Яковлева Людмила Андреевна	2020	<input checked="" type="checkbox"/>	1958-04-14	4
6	27	Доронина Анастасия Александровна	2011	<input checked="" type="checkbox"/>	1997-03-15	7

Рисунок 22 – До изменения

	id_employee	full_name_employee	seniority_begin	gender_employee	birth_day_employee	id_position
1	8	Цветков Александр Александрович	2016	<input type="checkbox"/>	1957-12-06	6
2	9	Мишин Алексей Викторович	2011	<input type="checkbox"/>	1973-09-19	3
3	11	Дубкова Людмила Андреевна	2015	<input checked="" type="checkbox"/>	1966-12-01	7
4	25	Цветков Алексей Сергеевич	2015	<input type="checkbox"/>	1995-09-25	5
5	26	Яковлева Людмила Андреевна	2020	<input checked="" type="checkbox"/>	1958-04-14	4
6	27	Доронина Анастасия Александровна	2011	<input checked="" type="checkbox"/>	1997-03-15	7
7	31	Носов Георгий Аркадьевич	2020	<input type="checkbox"/>	1995-05-12	7

Рисунок 23 – После изменения

14.Перевести сотрудника Мишина Алексея Викторовича с должности уборщика на должность охранника.

```
UPDATE emp_twenty_ten SET id_position = 6 WHERE
full_name_employee = 'Мишин Алексей Викторович';
```

	id_employee	full_name_employee	seniority_begin	gender_employee	birth_day_employee	id_position
1	8	Цветков Александр Александрович	2016	<input type="checkbox"/>	1957-12-06	6
2	9	Мишин Алексей Викторович	2011	<input type="checkbox"/>	1973-09-19	3
3	11	Дубкова Людмила Андреевна	2015	<input checked="" type="checkbox"/>	1966-12-01	7
4	25	Цветков Алексей Сергеевич	2015	<input type="checkbox"/>	1995-09-25	5
5	26	Яковлева Людмила Андреевна	2020	<input checked="" type="checkbox"/>	1958-04-14	4
6	27	Доронина Анастасия Александровна	2011	<input checked="" type="checkbox"/>	1997-03-15	7
7	31	Носов Георгий Аркадьевич	2020	<input type="checkbox"/>	1995-05-12	7

Рисунок 24 – До изменения

	id_employee	full_name_employee	seniority_begin	gender_employee	birth_day_employee	id_position
1	8	Цветков Александр Александрович	2016	<input type="checkbox"/>	1957-12-06	6
2	11	Дубкова Людмила Андреевна	2015	<input checked="" type="checkbox"/>	1966-12-01	7
3	25	Цветков Алексей Сергеевич	2015	<input type="checkbox"/>	1995-09-25	5
4	26	Яковлева Людмила Андреевна	2020	<input checked="" type="checkbox"/>	1958-04-14	4
5	27	Доронина Анастасия Александровна	2011	<input checked="" type="checkbox"/>	1997-03-15	7
6	31	Носов Георгий Аркадьевич	2020	<input type="checkbox"/>	1995-05-12	7
7	9	Мишин Алексей Викторович	2011	<input type="checkbox"/>	1973-09-19	6

Рисунок 25 – После изменения

15. Удалить всех сотрудников с ФИО Носов Георгий Аркадьевич.

```
DELETE FROM emp_twenty_ten WHERE full_name_employee = 'Носов
Георгий Аркадьевич';
```

	id_employee	full_name_employee	seniority_begin	gender_employee	birth_day_employee	id_position
1	8	Цветков Александр Александрович	2016	<input type="checkbox"/>	1957-12-06	6
2	11	Дубкова Людмила Андреевна	2015	<input checked="" type="checkbox"/>	1966-12-01	7
3	25	Цветков Алексей Сергеевич	2015	<input type="checkbox"/>	1995-09-25	5
4	26	Яковлева Людмила Андреевна	2020	<input checked="" type="checkbox"/>	1958-04-14	4
5	27	Доронина Анастасия Александровна	2011	<input checked="" type="checkbox"/>	1997-03-15	7
6	31	Носов Георгий Аркадьевич	2020	<input type="checkbox"/>	1995-05-12	7
7	9	Мишин Алексей Викторович	2011	<input type="checkbox"/>	1973-09-19	6

Рисунок 26 – До изменения

	id_employee	full_name_employee	seniority_begin	gender_employee	birth_day_employee	id_position
1	8	Цветков Александр Александрович	2016	<input type="checkbox"/>	1957-12-06	6
2	11	Дубкова Людмила Андреевна	2015	<input checked="" type="checkbox"/>	1966-12-01	7
3	25	Цветков Алексей Сергеевич	2015	<input type="checkbox"/>	1995-09-25	5
4	26	Яковлева Людмила Андреевна	2020	<input checked="" type="checkbox"/>	1958-04-14	4
5	27	Доронина Анастасия Александровна	2011	<input checked="" type="checkbox"/>	1997-03-15	7
6	9	Мишин Алексей Викторович	2011	<input type="checkbox"/>	1973-09-19	6

Рисунок 27 – После изменения

Функции SQL с базовыми типами (2 шт: 1 без аргументов, 1 с аргументами базовых типов)

1. Создать функцию, показывающую количество лет стажа у сотрудника к текущему моменту. Вывести ФИО сотрудника, год начала его стажа и срок стажа.

```

1.CREATE FUNCTION seniority_now(x integer) RETURNS integer AS $$
SELECT      (extract(year    from    current_date)    -    x)::integer;
$$ LANGUAGE SQL;

2.SELECT      full_name_employee,      seniority_begin,
seniority_now(seniority_begin) AS all_seniority FROM employee;

```

	full_name_employee	seniority_begin	all_seniority
1	Мишина Людмила Андреевна	2008	12
2	Яковлева Мария Александровна	2004	16
3	Кулагина Людмила Алексеевна	1993	27
4	Бобров Александр Викторович	1999	21
5	Дубкова Людмила Александровна	1985	35
6	Цветков Александр Александрович	2016	4
7	Одинцова Мария Алексеевна	1987	33
8	Дубкова Людмила Андреевна	2015	5
9	Дубков Алексей Михайлович	1994	26
10	Кулагин Виктор Александрович	2001	19
11	Кулагин Александр Викторович	1988	32
12	Дубков Виктор Александрович	1997	23
13	Одинцов Виктор Сергеевич	2002	18
14	Одинцова Анастасия Сергеевна	1997	23
15	Одинцова Мария Андреевна	2006	14
16	Доронин Сергей Викторович	1994	26
17	Боброва Евгения Алексеевна	2001	19
18	Дубкова Людмила Андреевна	1987	33
19	Кулагин Сергей Михайлович	1997	23
20	Доронина Евгения Андреевна	2002	18
21	Одинцов Виктор Александрович	1998	22
22	Цветков Алексей Сергеевич	2015	5
23	Яковлева Людмила Андреевна	2020	0
24	Доронина Анастасия Александровна	2011	9
25	Богданова Анастасия Андреевна	2001	19
26	Дубкова Анастасия Андреевна	1985	35
27	Доронин Александр Викторович	1987	33
28	Бобров Сергей Александрович	2001	19
29	Федосеева Евгения Андреевна	1993	27
30	Мишин Алексей Викторович	2011	9

Рисунок 28 – результат запроса

2. Создать функцию, показывающую название книги, чья стоимость минимальна. Вывести название книги, чья стоимость минимальна.

```

1.CREATE FUNCTION book_min() RETURNS varchar AS $$
SELECT title FROM book WHERE cost = (SELECT MIN(cost) FROM book);
$$ LANGUAGE SQL;

```

```
2.SELECT book_min();
```

	book_min
1	Стоя под радугой

Рисунок 29 – результат запроса

Функции SQL со сложными типами

1. Создать функцию, показывающую сколько стоила книга, если бы ее стоимость выросла в 1,5 раза. Вывести название книги, жанр, стоимость и стоимость, увеличенную в 1,5 раза.

```
1.CREATE FUNCTION double_cost(book) RETURNS numeric(11,2) AS $$
SELECT          $1.cost          *          1.5          AS          new_cost;
$$ LANGUAGE SQL;
```

```
2.SELECT title, genre, cost, double_cost(book.*) as doub_cost
FROM book;
```

	title	genre	cost	doub_cost
1	Вверх по лестнице	Драма	350.45	525.675
2	Одинокый мастер	Поэма	435.51	653.265
3	Вишня	Повесть	889.43	1334.145
4	Хороший доктор	Трагедия	811.72	1217.58
5	Сто лет одиночества	Роман	611.84	917.76
6	Стоя под радугой	Детектив	194.85	292.275
7	Замок из стекла	Трагедия	783.86	1175.79
8	Феи страны Фа	Фантастика	896.69	1345.035
9	Сияние	Трагедия	710.73	1066.095
10	Белый берег	Комедия	924.72	1387.08
11	Сладкая жизнь	Трагедия	667.87	1001.805
12	Жизнь в комнате	Сказка	501.69	752.535
13	Последний лист	Трагедия	584.51	876.765
14	Если пойдет снег	Поэма	714.23	1071.345
15	Дуняша	Комедия	1016.73	1525.095
16	Не время для драконов	Фантастика	554.23	831.345
17	Тайный лес	Рассказ	952.78	1429.17
18	Любовь и другие лекарства	Роман	686.81	1030.215
19	Остров сокровищ	Рассказ	466.45	699.675
20	Истинные намерения	Повесть	1011.28	1516.92
21	Море любви	Поэма	687.62	1031.43
22	Однажды днем	Повесть	1036.72	1555.08
23	Счастье и любовь	Поэма	586.85	880.275
24	Осенний мост	Фантастика	709.77	1064.655
25	Цветок жизни	Детектив	415.11	622.665
26	В движении	Роман	950.51	1425.765
27	Чиновник	Комедия	798.96	1198.44
28	Три билета	Трагедия	489.69	734.535
29	Дверь в лето	Рассказ	450.25	675.375
30	Объятия сна	Трагедия	930.26	1395.39

Рисунок 30 – результат запроса

Функции SQL с выходными параметрами

1. Создать функцию, показывающую количество дней, за которое книга была сдана и количество дней, за которое книга обязательно должна была быть сдана. Вывести дату выдачи, дату сдачи, обязательную дату сдачи и количество дней, за которое книга была сдана и за которое обязательно должна была быть сдана, если книга выдана начиная с 2015 года.

```
1. CREATE FUNCTION deli(x date, y date, z date, OUT back integer,
OUT
                        set_back
                        integer)
AS      'SELECT      (x      -      y),      (z      -      y) '
LANGUAGE SQL;
```

```
2.      SELECT      date_in,      date_return,      set_date_return,
deli(date_return, date_in, set_date_return) FROM delivery WHERE
date_return IS NOT NULL AND date_in > '2014-12-31';
```

	date_in	date_return	set_date_return	deli
1	2019-04-18	2019-05-12	2019-05-18	(24,30)
2	2019-09-24	2019-11-06	2019-11-09	(43,46)
3	2018-01-26	2018-02-19	2018-02-28	(24,33)
4	2017-02-22	2017-03-17	2017-03-20	(23,26)
5	2015-10-21	2015-11-19	2015-11-21	(29,31)
6	2018-10-30	2018-11-17	2018-12-05	(18,36)
7	2015-04-12	2015-04-27	2015-05-12	(15,30)

Рисунок 31 – результат запроса

Функции SQL с переменным числом аргументов

1. Создать функцию, вычисляющую раннюю дату из данных. Вывести id выдачи, дату выдачи, дату сдачи, обязательную дату сдачи и дату выдачи, вычисленную через функцию, для выдач после 2012 года.


```
1. CREATE FUNCTION in_del(VARIADIC arr date[]) RETURNS date AS $$
SELECT  min($1[i])    FROM    generate_subscripts($1, 1)    g(i);
$$ LANGUAGE SQL;
```

```
2. SELECT id_delivery, date_in, date_return, set_date_return,
in_del(date_in, date_return, set_date_return) AS func FROM
delivery WHERE date_in > '2012-12-31';
```

	id_delivery	date_in	date_return	set_date_return	func
1	1	2019-04-18	2019-05-12	2019-05-18	2019-04-18
2	7	2014-05-28	2014-06-18	2014-07-14	2014-05-28
3	8	2019-09-24	2019-11-06	2019-11-09	2019-09-24
4	12	2020-01-11	<null>	2020-02-13	2020-01-11
5	14	2018-01-26	2018-02-19	2018-02-28	2018-01-26
6	15	2019-11-09	<null>	2019-12-20	2019-11-09
7	16	2017-02-22	2017-03-17	2017-03-20	2017-02-22
8	17	2015-10-21	2015-11-19	2015-11-21	2015-10-21
9	21	2019-08-14	<null>	2019-09-14	2019-08-14
10	24	2013-10-21	2013-11-12	2013-11-13	2013-10-21
11	25	2018-01-23	<null>	2018-02-23	2018-01-23
12	26	2018-10-30	2018-11-17	2018-12-05	2018-10-30
13	27	2015-04-12	2015-04-27	2015-05-12	2015-04-12

Рисунок 32 – результат запроса

Функции SQL со значениями аргументов по умолчанию

1. Создать функцию, которая бы увеличивала зарплату сотрудника на введенное количество условных единиц. Иначе, зарплата бы увеличивалась на 3500. Вывести ФИО сотрудника, название его должности, зарплату на данный момент и зарплату, которая могла бы быть с использованием функции.

```
1. CREATE FUNCTION more_salary(a numeric(11,2), b numeric(11,2)
DEFAULT                                     3500)
RETURNS                                numeric(11,2)          AS
SELECT $1 + $2;
```

```
$$ LANGUAGE SQL;
```

```
2. SELECT      full_name_employee,      title_position,      salary,
      more_salary(salary, 5000) FROM employee, position WHERE
      employee.id_position = position.id_position;
```

```
3. SELECT      full_name_employee,      title_position,      salary,
      more_salary(salary)      FROM      employee,      position      WHERE
      employee.id_position = position.id_position;
```

	full_name_employee	title_position	salary	more_salary
1	Мишина Людмила Андреевна	Заведующий	38247.35	43247.35
2	Яковлева Мария Александровна	Системный администратор	35976.96	40976.96
3	Кулагина Людмила Алексеевна	Системный администратор	35976.96	40976.96
4	Бобров Александр Викторович	Библиотекарь	29374.98	34374.98
5	Дубкова Людмила Александровна	Библиотекарь	29374.98	34374.98
6	Цветков Александр Александрович	Охранник	22317.31	27317.31
7	Одинцова Мария Алексеевна	Библиотекарь	29374.98	34374.98
8	Дубкова Людмила Андреевна	Системный администратор	35976.96	40976.96
9	Дубков Алексей Михайлович	Охранник	22317.31	27317.31
10	Кулагин Виктор Александрович	Уборщик	21144.92	26144.92
11	Кулагин Александр Викторович	Библиотекарь	29374.98	34374.98
12	Дубков Виктор Александрович	Библиотекарь	29374.98	34374.98
13	Одинцов Виктор Сергеевич	Бухгалтер	37721.39	42721.39
14	Одинцова Анастасия Сергеевна	Системный администратор	35976.96	40976.96
15	Одинцова Мария Андреевна	Уборщик	21144.92	26144.92
16	Доронин Сергей Викторович	Бухгалтер	37721.39	42721.39
17	Боброва Евгения Алексеевна	Библиотекарь	29374.98	34374.98
18	Дубкова Людмила Андреевна	Бухгалтер	37721.39	42721.39
19	Кулагин Сергей Михайлович	Системный администратор	35976.96	40976.96
20	Доронина Евгения Андреевна	Библиотекарь	29374.98	34374.98
21	Одинцов Виктор Александрович	Библиотекарь	29374.98	34374.98
22	Цветков Алексей Сергеевич	Библиотекарь	29374.98	34374.98
23	Яковлева Людмила Андреевна	Бухгалтер	37721.39	42721.39
24	Доронина Анастасия Александровна	Системный администратор	35976.96	40976.96
25	Богданова Анастасия Андреевна	Бухгалтер	37721.39	42721.39
26	Дубкова Анастасия Андреевна	Уборщик	21144.92	26144.92
27	Доронин Александр Викторович	Уборщик	21144.92	26144.92
28	Бобров Сергей Александрович	Директор	43500.25	48500.25
29	Федосеева Евгения Андреевна	Уборщик	21144.92	26144.92
30	Мишин Алексей Викторович	Охранник	22317.31	27317.31

Рисунок 33 – результат запроса со введенным аргументом

	full_name_employee	title_position	salary	more_salary
1	Мишина Людмила Андреевна	Заведующий	38247.35	41747.35
2	Яковлева Мария Александровна	Системный администратор	35976.96	39476.96
3	Кулагина Людмила Алексеевна	Системный администратор	35976.96	39476.96
4	Бобров Александр Викторович	Библиотекарь	29374.98	32874.98
5	Дубкова Людмила Александровна	Библиотекарь	29374.98	32874.98
6	Цветков Александр Александрович	Охранник	22317.31	25817.31
7	Одинцова Мария Алексеевна	Библиотекарь	29374.98	32874.98
8	Дубкова Людмила Андреевна	Системный администратор	35976.96	39476.96
9	Дубков Алексей Михайлович	Охранник	22317.31	25817.31
10	Кулагин Виктор Александрович	Уборщик	21144.92	24644.92
11	Кулагин Александр Викторович	Библиотекарь	29374.98	32874.98
12	Дубков Виктор Александрович	Библиотекарь	29374.98	32874.98
13	Одинцов Виктор Сергеевич	Бухгалтер	37721.39	41221.39
14	Одинцова Анастасия Сергеевна	Системный администратор	35976.96	39476.96
15	Одинцова Мария Андреевна	Уборщик	21144.92	24644.92
16	Доронин Сергей Викторович	Бухгалтер	37721.39	41221.39
17	Боброва Евгения Алексеевна	Библиотекарь	29374.98	32874.98
18	Дубкова Людмила Андреевна	Бухгалтер	37721.39	41221.39
19	Кулагин Сергей Михайлович	Системный администратор	35976.96	39476.96
20	Доронина Евгения Андреевна	Библиотекарь	29374.98	32874.98
21	Одинцов Виктор Александрович	Библиотекарь	29374.98	32874.98
22	Цветков Алексей Сергеевич	Библиотекарь	29374.98	32874.98
23	Яковлева Людмила Андреевна	Бухгалтер	37721.39	41221.39
24	Доронина Анастасия Александровна	Системный администратор	35976.96	39476.96
25	Богданова Анастасия Андреевна	Бухгалтер	37721.39	41221.39
26	Дубкова Анастасия Андреевна	Уборщик	21144.92	24644.92
27	Доронин Александр Викторович	Уборщик	21144.92	24644.92
28	Бобров Сергей Александрович	Директор	43500.25	47000.25
29	Федосеева Евгения Андреевна	Уборщик	21144.92	24644.92
30	Мишин Алексей Викторович	Охранник	22317.31	25817.31

Рисунок 34 – результат запроса с аргументом по умолчанию

Функции SQL, возвращающие таблицы (TABLE)

1. Создать функцию, которая выводила id выдачи, ФИО читателя, не сдавшего книгу, и количество дней, на которое он просрочил книгу на конкретную указанную дату. Вывести id выдачи, ФИО читателя, не сдавшего книгу, и количество дней, на которое он просрочил книгу на 2020-02-25.

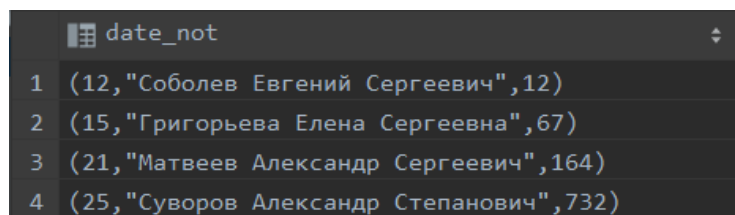
```

1.CREATE FUNCTION date_not (x date)
RETURNS TABLE(id_delivery integer, full_name_reader varchar,
num_day integer) AS $$
SELECT id_delivery, full_name_reader , ($1 - set_date_return)

```

```
FROM reader, delivery WHERE reader.id_reader = delivery.id_reader
AND date_return IS NULL AND $1 > set_date_return;
$$ LANGUAGE SQL;
```

```
2.SELECT date_not('2020-02-25');
```



	date_not
1	(12,"Соболев Евгений Сергеевич",12)
2	(15,"Григорьева Елена Сергеевна",67)
3	(21,"Матвеев Александр Сергеевич",164)
4	(25,"Суворов Александр Степанович",732)

Рисунок 35 – результат запроса

Функция PL/pgSQL

1. Создать функцию, которая выводила бы 3 книги с минимальной стоимостью в заданном диапазоне. Вывести название и стоимость самых дешевых книг в заданном диапазоне

```
1. CREATE FUNCTION min_three_book(cost_a numeric (11,2), cost_b
numeric(11,2)) RETURNS setof book AS $$
```

```
DECLARE
```

```
    r book%rowtype;
```

```
    count integer DEFAULT 0;
```

```
    pred numeric(11,2) DEFAULT 0.0;
```

```
BEGIN
```

```
FOR r IN
```

```
select * from book WHERE cost BETWEEN cost_a AND cost_b
ORDER BY cost
```

```
LOOP
```

```

        IF r.cost != pred THEN

            count = count + 1;

        END IF;

        IF COUNT = 4 THEN

            RETURN;

        END IF;

        RETURN next r;

    END LOOP;

END;

$$ LANGUAGE plpgsql;

2. SELECT title, cost FROM min_three_book(400, 800);

```

	title	cost
1	Цветок жизни	415.11
2	Одинокий мастер	435.51
3	Дверь в лето	450.25

Рисунок 36 – результат запроса

Два триггера для INSERT / UPDATE / DELETE (разные комбинации с BEFORE, AFTER и INSTEAD OF)

1. Создать триггер, который перед добавлением записи в таблицу сотрудник проверял бы, все ли заполнены поля, и ставил бы в год начала стажа текущий год.

```

1. CREATE FUNCTION emp_check() RETURNS trigger AS $emp_check$

```

```

BEGIN

IF NEW.full_name_employee IS NULL THEN

RAISE EXCEPTION 'employee_name cannot be null';

END IF;

IF NEW.birth_day_employee IS NULL THEN

RAISE EXCEPTION '%birthday cannot be null',
NEW.full_name_employee;

END IF;

IF NEW.id_position IS NULL THEN

RAISE EXCEPTION '%employee must hold a position',
NEW.full_name_employee;

END IF;

IF NEW.seniority_begin IS NULL THEN

NEW.seniority_begin := extract(year from current_date);

END IF;

RETURN NEW;

END;

$emp_check$ LANGUAGE plpgsql;

```

2. CREATE TRIGGER emp_check BEFORE INSERT ON employee

FOR EACH ROW EXECUTE FUNCTION emp_check();

3. INSERT INTO employee (seniority_begin, birth_day_employee, gender_employee, id_position) VALUES (2015, '1995-05-05', false, 5);

```
[2020-06-11 09:40:26] [P0001] ERROR: employee_name cannot be null
[2020-06-11 09:40:26] Где: PL/pgSQL function emp_check() line 5 at RAISE
```

Рисунок 37 – результат запроса

```
4. INSERT INTO employee (full_name_employee, birth_day_employee,
gender_employee, id_position) VALUES ('Митин Георгий
Олегович', '1995-05-05', false, 5);
```

31	34 Митин Георгий Олегович	2020		1995-05-05	5
----	---------------------------	------	--	------------	---

Рисунок 38 – результат запроса, когда с помощью триггера добавился
год начала стажа

2. Создать триггер, который после того, как читатель был добавлен, добавлял его ФИО и дату рождения в таблицу с архивом новых читателей.

```
1. CREATE TABLE reader_new (
    reader_name varchar,
    birth_reader date
);

2. CREATE FUNCTION reader_new() RETURNS trigger AS $reader_new$
BEGIN
    INSERT INTO reader_new VALUES (NEW.full_name_reader,
    NEW.birth_day_reader);
    RETURN NEW;
END;
```

```

$reader_new$ LANGUAGE plpgsql;

3. CREATE TRIGGER reader_new AFTER INSERT ON reader

    FOR EACH ROW EXECUTE FUNCTION reader_new();

4. INSERT INTO reader (full_name_reader, birth_day_reader) VALUES
    ('Слонова Мария Игоревна', '2005-12-12');

5. SELECT * FROM reader_new;

```

	reader_name	birth_reader
1	Слонова Мария Игоревна	2005-12-12

Рисунок 39 – результат запроса

3. Создать триггер, который бы перед тем, как удалить сотрудника, добавлял его ФИО и дату рождения в таблицу с архивом сотрудников.

```

1. CREATE TABLE archive_emp (

    emp_name varchar,

    birth_day date

);

2. CREATE FUNCTION emp_arch() RETURNS trigger AS $emp_arch$

BEGIN

    INSERT INTO archive_emp VALUES (OLD.full_name_employee,

    OLD.birth_day_employee);

    RETURN OLD;

END;

```



```

$emp_arch$ LANGUAGE plpgsql;

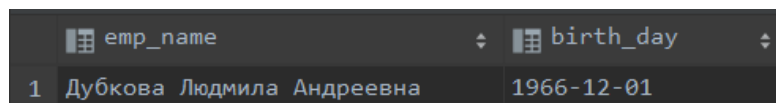
3. CREATE TRIGGER emp_arch BEFORE DELETE ON employee

    FOR EACH ROW EXECUTE FUNCTION emp_arch();

4. DELETE FROM employee WHERE id_employee = 11;

5. SELECT * FROM archive_emp;

```



	emp_name	birth_day
1	Дубкова Людмила Андреевна	1966-12-01

Рисунок 40 – результат запроса

4. Создать триггер, который бы после удаления читателя выводил бы сообщение «deleted successfully».

```

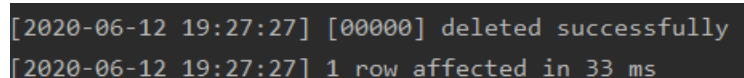
1. CREATE FUNCTION reader_del() RETURNS trigger AS $reader_del$
    BEGIN
        RAISE          NOTICE          'deleted          successfully';
        RETURN                                OLD;
    END;
    $reader_del$ LANGUAGE plpgsql;

2. CREATE TRIGGER reader_del AFTER DELETE ON reader

    FOR EACH ROW EXECUTE FUNCTION reader_del();

3. DELETE FROM reader WHERE id_reader = 11;

```



```

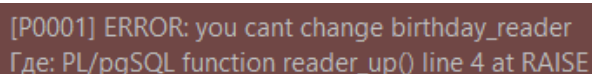
[2020-06-12 19:27:27] [00000] deleted successfully
[2020-06-12 19:27:27] 1 row affected in 33 ms

```

Рисунок 41 – результат запроса

5. Создать триггер, который бы перед изменением читателя выдавал ошибку, если бы пытались изменить дату рождения.

```
1. CREATE FUNCTION reader_up() RETURNS trigger AS $reader_up$  
  
    BEGIN  
  
    IF NEW.birth_day_reader != OLD.birth_day_reader THEN  
  
        RAISE EXCEPTION 'you cant change birthday_reader';  
  
    END IF;  
  
    END;  
  
$reader_up$ LANGUAGE plpgsql;  
  
2. CREATE TRIGGER reader_up BEFORE UPDATE ON reader  
  
    FOR EACH ROW EXECUTE FUNCTION reader_up();  
  
3. CREATE TRIGGER arch_book INSTEAD OF UPDATE ON tragedy  
  
    FOR EACH ROW EXECUTE FUNCTION arch_book();  
  
4. UPDATE reader SET birth_day_reader = '2005-11-20' WHERE  
    id_reader = 2;
```



[P0001] ERROR: you cant change birthday_reader
Где: PL/pgSQL function reader_up() line 4 at RAISE

Рисунок 42 – результат запроса

6. Создать триггер, который бы вместо того, чтобы изменять книгу, добавлял бы в архив текущую дату попытки изменения.

```
1. CREATE TABLE arch_book_upd (  
  
    date_change date);
```

```

2. CREATE FUNCTION arch_book() RETURNS trigger AS $arch_book$

BEGIN

INSERT INTO arch_book_upd VALUES (CURRENT_DATE);

RETURN OLD;

END;

$arch_book$ LANGUAGE plpgsql;

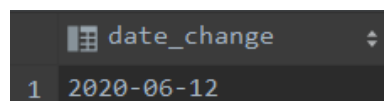
3. CREATE TRIGGER arch_book INSTEAD OF UPDATE ON tragedy

FOR EACH ROW EXECUTE FUNCTION arch_book();

6. UPDATE tragedy SET cost = 450 WHERE ISBN = 5;

7. SELECT * FROM arch_book_upd;

```



	date_change
1	2020-06-12

Рисунок 43 – результат запроса

Выводы

В ходе данной лабораторной работы были получены навыки написания представлений, функций SQL с базовыми и сложными типами, выходными параметрами. Научилась создавать функции PL/pgSQL и триггеры.