

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Базы данных»**  
**Тема: Реализация базы данных в СУБД PostgreSQL**

Студент гр. 1303

\_\_\_\_\_

Смирнов Д.Ю.

Преподаватель

\_\_\_\_\_

Заславский М.М.

Санкт-Петербург

2023

## **Цель работы**

Развернуть локально PostgreSQL, написать запросы для создания и заполнения таблиц, написать запросы к БД, отвечающие на вопросы задания.

## **Задание**

### Вариант 19

Пусть требуется создать программную систему, предназначенную для работников технического архива предприятия. Технический архив содержит стеллажи, полки и ячейки, в которых хранится документация. Ячейка архива может быть пустой или хранить все экземпляры одного документа. Каждый экземпляр документации имеет инвентарный номер и название. В базе данных должна храниться следующая информация о каждом документе архива: номер стеллажа, номер полки, номер ячейки, где хранится документ, название документа и название темы, к которой он относится, его инвентарный номер, количество экземпляров документа, содержащихся в ячейке, дата поступления документа в архив. Документ может быть востребован абонентом архива. Абонент характеризуется фамилией, именем, отчеством, номером и телефоном отдела, где он работает. Работники архива, выдавая документ, должны зафиксировать, когда и кому он был выдан. Архив может пополняться документами, как новыми, так и копиями уже имеющихся в архиве. Экземпляр документа может быть утрачен. Возможна закупка новых стеллажей и списание старых. Документ может менять место хранения и инвентарный номер. Возможно и изменение сведений об абонентах. Абонент может менять фамилию, перейти в другой отдел, уволится с предприятия. Возможно изменение номеров телефонов отделов. Работнику архива могут потребоваться следующие сведения:

- Название наиболее востребованного документа?
- Общее количество документов на заданную тему?
- Тема документа по заданному названию?

- Название документа, который имеется в архиве в максимальном количестве экземпляров?
- Фамилия, имя и отчество абонента, который брал указанный документ последним?
- Есть ли в архиве пустые стеллажи, полки, ячейки, и в каком количестве?
- Список документов, не востребованных в течение более, чем 5 лет?

## Выполнение работы

Создана база данных под названием «Архив», рисунок 1.

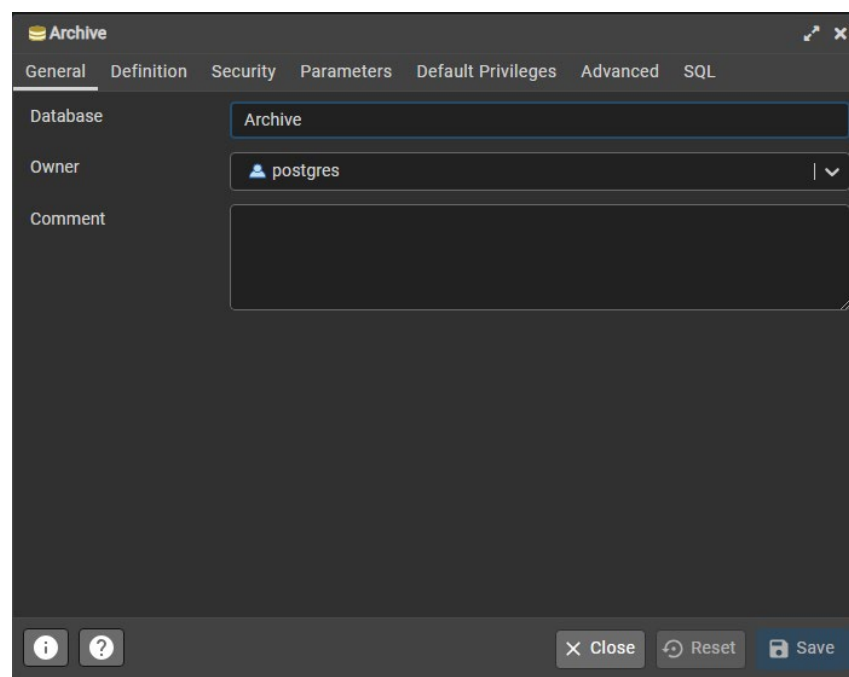


Рисунок 1 – Создание БД "Архив"

Созданы таблицы «*shelf*», «*rack*», «*cell*», «*instance*», «*department*», «*abonent*», «*issues*», «*issues\_archive*» в соответствии со структурой БД, а также тип перечисление «*instance\_status*». Запросы на создание представлены на рисунках 2-3.

Исходный код всех запросов представлен в приложении Б.

```

1 CREATE TYPE instance_status AS ENUM('есть', 'утерян');
2
3 CREATE TABLE shelf(
4     shelf_id SERIAL PRIMARY KEY,
5     unique_shelf_number INTEGER UNIQUE CHECK(unique_shelf_number > 0) NOT NULL
6 );
7
8 CREATE TABLE rack(
9     rack_id SERIAL PRIMARY KEY,
10    shelf_id INTEGER NOT NULL,
11    unique_rack_number INTEGER UNIQUE CHECK(unique_rack_number > 0) NOT NULL,
12    FOREIGN KEY (shelf_id) REFERENCES shelf(shelf_id) ON DELETE RESTRICT
13 );
14
15 CREATE TABLE cell(
16     cell_id SERIAL PRIMARY KEY,
17     rack_id INTEGER NOT NULL,
18     unique_cell_number INTEGER UNIQUE CHECK(unique_cell_number > 0) NOT NULL,
19     FOREIGN KEY (rack_id) REFERENCES rack(rack_id) ON DELETE RESTRICT
20 );
21
22 CREATE TABLE document(
23     document_id SERIAL PRIMARY KEY,
24     theme_name VARCHAR(100) NOT NULL,
25     document_title VARCHAR(250) NOT NULL,
26     inventory_number VARCHAR(14) UNIQUE NOT NULL,
27     cell_id INTEGER NOT NULL,
28     FOREIGN KEY (cell_id) REFERENCES cell(cell_id) ON DELETE RESTRICT
29 );

```

Рисунок 2 – запросы на создание, часть 1

```

1 CREATE TABLE instance(
2     instance_id SERIAL PRIMARY KEY,
3     inventory_number VARCHAR(14) UNIQUE NOT NULL,
4     status instance_status NOT NULL,
5     title VARCHAR(250) NOT NULL,
6     document_id INTEGER NOT NULL,
7     FOREIGN KEY (document_id) REFERENCES document(document_id) ON DELETE CASCADE
8 );
9
10 CREATE TABLE department(
11     department_id SERIAL PRIMARY KEY,
12     department_name VARCHAR(100) NOT NULL,
13     department_number VARCHAR(50) NOT NULL
14 );
15
16 CREATE TABLE abonent(
17     abonent_id SERIAL PRIMARY KEY,
18     phone_number VARCHAR(50) NOT NULL,
19     name VARCHAR(100) NOT NULL,
20     surname VARCHAR(100),
21     middle_name VARCHAR(100),
22     department_id INTEGER NOT NULL,
23     FOREIGN KEY (department_id) REFERENCES department(department_id) ON DELETE CASCADE
24 );
25
26 CREATE TABLE issues(
27     instance_id INTEGER,
28     abonent_id INTEGER,
29     PRIMARY KEY(instance_id, abonent_id),
30     date_of_issue DATE NOT NULL,
31     FOREIGN KEY (instance_id) REFERENCES instance(instance_id) ON DELETE CASCADE,
32     FOREIGN KEY (abonent_id) REFERENCES abonent(abonent_id) ON DELETE CASCADE
33 );
34
35 CREATE TABLE issues_archive(
36     issues_archive_id SERIAL PRIMARY KEY,
37     document_id INTEGER NOT NULL,
38     date_of_issue DATE NOT NULL,
39     name VARCHAR(100) NOT NULL,
40     surname VARCHAR(100),
41     middle_name VARCHAR(100),
42     FOREIGN KEY (document_id) REFERENCES document(document_id) ON DELETE CASCADE
43 );

```

Рисунок 3 – запросы на создание, часть 2

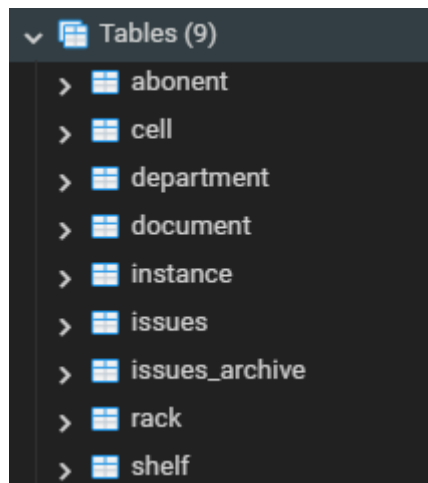


Рисунок 4 – Полученные таблицы

Запросы на заполнение таблиц представлены на рисунках 5-6.

```

1  INSERT INTO shelf(unique_shelf_number) VALUES
2      (1789131),
3      (2124789),
4      (3247893),
5      (4489534),
6      (5526783),
7      (6890146);
8  INSERT INTO rack(shelf_id, unique_rack_number) VALUES
9      (1, 1247890),
10     (1, 1589161),
11     (2, 2781313),
12     (2, 2894131),
13     (3, 3471123),
14     (3, 3857841),
15     (4, 4938732),
16     (4, 4538713),
17     (5, 5982412),
18     (5, 5001378),
19     (6, 6982412),
20     (6, 6001378);
21  INSERT INTO cell(rack_id, unique_cell_number) VALUES
22     (1, 1189313),
23     (1, 1289431),
24     (2, 2178941),
25     (2, 2278914),
26     (3, 3178913),
27     (3, 3289234),
28     (4, 4138732),
29     (5, 5182412),
30     (6, 6182412),
31     (7, 7178123),
32     (8, 8189723),
33     (9, 9136781),
34     (10, 8901331),
35     (11, 1678921),
36     (12, 8913124);

```

Рисунок 5 – заполнение таблиц, часть 1

```

37 INSERT INTO document(theme_name, document_title, inventory_number, cell_id) VALUES
38 ('учет инвентаря', 'Учетный инвентарь за 2023', '0100000012023', 1),
39 ('документация', 'Правила инвентаризации', '01000100012023', 4),
40 ('отчеты экономистов', 'План экономического развития на 2024', '02000020242023', 5),
41 ('налоговая', 'Результаты налоговой инспекции за 2022', '02000011112022', 7),
42 ('документация', 'Комплектная документация на станки от Bosh', '03000019982019', 8),
43 ('история предприятия', 'История предприятия с момента октябрьской революции', '07108104522010', 9),
44 ('охрана труда', 'Политика организации по охране труда', '08000035642010', 10);
45 INSERT INTO instance(title, inventory_number, status, document_id) VALUES
46 ('Ученные станки', '000000010001', 'есть', 1),
47 ('Правила инвентаризации экз.1', '000100010001', 'есть', 2),
48 ('Правила инвентаризации экз.2', '000100010002', 'есть', 2),
49 ('Новый экономический план на предприятии', '000020240001', 'есть', 3),
50 ('Документация на станок BOSH №313ADC31', '000019980001', 'есть', 5);
51 INSERT INTO department(department_name, department_number) VALUES
52 ('материально-технического снабжения', '013-00-06'),
53 ('планово-экономический', '916-14-61'),
54 ('конструкторский', '343-74-62'),
55 ('управления персоналом', '949-25-48'),
56 ('военизированная охрана', '045-22-26');
57 INSERT INTO abonent(phone_number, surname, name, middle_name, department_id) VALUES
58 ('7(416)127-21-37', 'Осипова', 'Мария', 'Дмитриевна', 1),
59 ('7(589)183-65-03', 'Кириллов', 'Артём', 'Робертович', 1),
60 ('7(842)993-16-33', 'Богданова', 'Эмилия', 'Александровна', 2),
61 ('7(712)011-66-67', 'Кузнецов', 'Фёдор', 'Никитич', 3),
62 ('7(404)450-48-49', 'Буров', 'Илья', 'Лукич', 5);
63 INSERT INTO issues(instance_id, abonent_id, date_of_issue) VALUES
64 (1, 4, '2023-09-01'),
65 (5, 4, '2023-10-03'),
66 (2, 1, '2023-10-12'),
67 (3, 2, '2023-10-15'),
68 (4, 5, '2023-10-20');
69 INSERT INTO issues_archive(document_id, date_of_issue, surname, name, middle_name) VALUES
70 (6, '2010-01-01', 'Робертов', 'Артём', 'Данилович'),
71 (7, '2013-10-18', 'Соколова', 'Софья', 'Тимуровна'),
72 (5, '2020-11-11', 'Мухин', 'Антон', 'Михайлович'),
73 (5, '2022-07-15', 'Богданова', 'Эмилия', 'Александровна'),
74 (5, '2016-09-24', 'Буров', 'Илья', 'Лукич');

```

Рисунок 6 – заполнение таблиц, часть 2

Написаны запросы для работника архива:

1. Название наиболее востребованного документа, представлен на рисунке 7.

```

1  -- Название наиболее востребованного документа?
2  WITH all_issues AS (
3      SELECT document_id, date_of_issue FROM issues
4          INNER JOIN instance ON instance.instance_id = issues.instance_id
5      UNION ALL
6      SELECT document_id, date_of_issue FROM issues_archive
7  )
8  SELECT document_title, COUNT(*) AS Количество_выдач FROM all_issues
9      INNER JOIN document ON all_issues.document_id = document.document_id
10     GROUP BY document.document_id
11     HAVING COUNT(*) = (SELECT COUNT(*) FROM all_issues GROUP BY document_id ORDER BY 1 DESC LIMIT 1);

```

document_title	Количество_выдач
Комплектная документация на станки от Bosh	4

Рисунок 7 – наиболее востребованный документ

2. Количество документов на заданную тему, представлен на рисунке 8.

1	-- Общее количество документов на заданную тему?
2	SELECT COUNT(*) AS Количество FROM document
3	WHERE theme_name = 'документация';
Data Output Messages Notifications	
<div> <div>☰</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div>	
	Количество bigint
1	2

Рисунок 8 — количество документов на заданную тему

3. Тема документа по указанному названию, представлен на рисунке 9.

Query	Query History
1	-- Тема документа по заданному названию?
2	SELECT theme_name AS Тема FROM document
3	WHERE document_title = 'План экономического развития на 2024';
Data Output Messages Notifications	
<div> <div>☰</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div>	
	Тема character varying (100)
1	отчеты экономистов

Рисунок 9 - тема документа по названию

4. Название документа, который имеется в архиве в максимальном количестве экземпляров, представлен на рисунке 10.

Query	Query History
1	-- Название документа, который имеется в архиве в максимальном количестве экземпляров?
2	SELECT document.document_id, document_title AS Название FROM document
3	INNER JOIN instance ON document.document_id = instance.document_id
4	GROUP BY document.document_id
5	HAVING COUNT(*) = (SELECT COUNT(*) FROM instance
6	WHERE status <> 'утерян'
7	GROUP BY document_id
8	ORDER BY 1 DESC
9	LIMIT 1);
Data Output Messages Notifications	
<div> <div>☰</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div>	
	document_id [PK] integer
	Название character varying (250)
1	2 Правила инвентаризации

Рисунок 10 — название документа, у которого максимальное количество экземпляров

5. ФИО абонента, который брал указанный документ последним, представлен на рисунке 11.

```

1  -- Фамилия, имя и отчество абонента, который брал указанный документ последним?
2  SELECT фамилия, имя, отчество FROM (
3  SELECT surname AS Фамилия, name AS Имя, middle_name AS Отчество, date_of_issue, document_id FROM issues_archive
4  UNION ALL
5  SELECT surname AS Фамилия, name AS Имя, middle_name AS Отчество, date_of_issue, document_id FROM issues
6      INNER JOIN abonent ON abonent.abonent_id = issues.abonent_id
7      INNER JOIN instance ON instance.instance_id = issues.instance_id
8  ) AS takers_info
9  WHERE document_id = (SELECT document_id FROM document WHERE document_title='Политика организации по охране труда')
10 ORDER BY date_of_issue DESC
11 LIMIT 1;

```

Фамилия character varying (100)	Имя character varying (100)	Отчество character varying (100)
1	Соколова	Софья Тимуровна

Рисунок 11 – ФИО абонента, бравшего указанный документ последним

6. Есть ли в архиве пустые ячейки, полки, стеллажи и в каком количестве, представлен на рисунке 12.

```

1  -- Есть ли в архиве пустые стеллажи, полки, ячейки, и в каком количестве?
2  WITH free_cells AS (
3  SELECT cell.cell_id, rack.rack_id, shelf.shelf_id, CASE WHEN document_id IS NULL THEN 1 ELSE 0 END AS cell_empty FROM cell
4      INNER JOIN rack ON rack.rack_id = cell.rack_id
5      INNER JOIN shelf ON shelf.shelf_id = rack.shelf_id
6      LEFT JOIN document ON document.cell_id = cell.cell_id
7  ),
8  free_racks AS (
9      SELECT rack_id FROM free_cells
10     GROUP BY rack_id
11     HAVING COUNT(*) = SUM(cell_empty)
12 ),
13 free_shelves AS (
14     SELECT shelf_id FROM free_cells
15     GROUP BY shelf_id
16     HAVING COUNT(*) = SUM(cell_empty)
17 )
18 SELECT 'Ячеек' AS Название, SUM(cell_empty) AS Количество FROM free_cells
19 UNION ALL
20 SELECT 'Полок' AS Название, COUNT(rack_id) AS Количество FROM free_racks
21 UNION ALL
22 SELECT 'Стеллажей' AS Название, COUNT(shelf_id) AS Количество FROM free_shelves;

```

Data Output

Messages

Notifications

Название

Количество

text

bigint

1

Ячеек

8

2

Полок

5

3

Стеллажей

2

Рисунок 12 – запрос на пустые ячейки, полки, стеллажи

7. Список документов, не востребованных более чем 5 лет, представлен на рисунке 13.

```

1  -- Список документов, не востребованных в течение более, чем 5 лет?
2  WITH all_issues AS (
3      SELECT document_id, date_of_issue FROM issues
4      INNER JOIN instance ON instance.instance_id = issues.instance_id
5      UNION ALL
6      SELECT document_id, date_of_issue FROM issues_archive
7  )
8  SELECT document.document_id, document_title, theme_name, MAX(date_of_issue) AS Дата_последней_выдачи FROM document
9  LEFT JOIN all_issues ON document.document_id = all_issues.document_id
10 GROUP BY document.document_id
11 HAVING MAX(date_of_issue) IS NULL OR age(MAX(date_of_issue)) > interval '5 years';

```

document_id [PK] integer	document_title character varying (250)	theme_name character varying (100)	Дата_последней_выдачи date
1	4	Результаты налоговой инспекции за 2022	налоговая
2	6	История предприятия с момента октябрьской революц...	история предприятия
3	7	Политика организации по охране труда	охрана труда

Рисунок 13 – документы, не востребованные более чем 5 лет



**Вывод.**

В ходе выполнения работы был развернут локально PostgreSQL Server, написаны запросы для создания и заполнения таблиц в соответствии со структурой БД, получен опыт написания различных запросов к БД.

## **ПРИЛОЖЕНИЕ А**

### **ССЫЛКИ**

Pull Request: <https://github.com/moevm/sql-2023-1303/pull/33>

DB Fiddle: <https://www.db-fiddle.com/f/4aeZHHq6yKf4YRBujHyDjB/73>

## ПРИЛОЖЕНИЕ Б

### ИСХОДНЫЙ КОД

Файл lab2.sql:

```
CREATE TYPE instance_status AS ENUM('есть', 'утерян');

CREATE TABLE shelf(
    shelf_id SERIAL PRIMARY KEY,
    unique_shelf_number INTEGER UNIQUE CHECK(unique_shelf_number > 0) NOT
NULL
);

CREATE TABLE rack(
    rack_id SERIAL PRIMARY KEY,
    shelf_id INTEGER NOT NULL,
    unique_rack_number INTEGER UNIQUE CHECK(unique_rack_number > 0) NOT
NULL,
    FOREIGN KEY (shelf_id) REFERENCES shelf(shelf_id) ON DELETE RESTRICT
);

CREATE TABLE cell(
    cell_id SERIAL PRIMARY KEY,
    rack_id INTEGER NOT NULL,
    unique_cell_number INTEGER UNIQUE CHECK(unique_cell_number > 0) NOT
NULL,
    FOREIGN KEY (rack_id) REFERENCES rack(rack_id) ON DELETE RESTRICT
);

CREATE TABLE document(
    document_id SERIAL PRIMARY KEY,
    theme_name VARCHAR(100) NOT NULL,
    document_title VARCHAR(250) NOT NULL,
    inventory_number VARCHAR(14) UNIQUE NOT NULL,
    cell_id INTEGER NOT NULL,
    FOREIGN KEY (cell_id) REFERENCES cell(cell_id) ON DELETE RESTRICT
);

CREATE TABLE instance(
    instance_id SERIAL PRIMARY KEY,
    inventory_number VARCHAR(14) UNIQUE NOT NULL,
    status instance_status NOT NULL,
    title VARCHAR(250) NOT NULL,
    document_id INTEGER NOT NULL,
    FOREIGN KEY (document_id) REFERENCES document(document_id) ON DELETE
CASCADE
);

CREATE TABLE department(
    department_id SERIAL PRIMARY KEY,
    department_name VARCHAR(100) NOT NULL,
    department_number VARCHAR(50) NOT NULL
);

CREATE TABLE abonent(
    abonent_id SERIAL PRIMARY KEY,
```

```

        phone_number VARCHAR(50) NOT NULL,
        name VARCHAR(100) NOT NULL,
        surname VARCHAR(100),
        middle_name VARCHAR(100),
        department_id INTEGER NOT NULL,
        FOREIGN KEY (department_id) REFERENCES department(department_id) ON
DELETE CASCADE
);

CREATE TABLE issues(
    instance_id INTEGER,
    abonent_id INTEGER,
    PRIMARY KEY(instance_id, abonent_id),
    date_of_issue DATE NOT NULL,
    FOREIGN KEY (instance_id) REFERENCES instance(instance_id) ON DELETE
CASCADE,
    FOREIGN KEY (abonent_id) REFERENCES abonent(abonent_id) ON DELETE
CASCADE
);

CREATE TABLE issues_archive(
    issues_archive_id SERIAL PRIMARY KEY,
    document_id INTEGER NOT NULL,
    date_of_issue DATE NOT NULL,
    name VARCHAR(100) NOT NULL,
    surname VARCHAR(100),
    middle_name VARCHAR(100),
    FOREIGN KEY (document_id) REFERENCES document(document_id) ON DELETE
CASCADE
);

INSERT INTO shelf(unique_shelf_number) VALUES
    (1789131),
    (2124789),
    (3247893),
    (4489534),
    (5526783),
    (6890146);

INSERT INTO rack(shelf_id, unique_rack_number) VALUES
    (1, 1247890),
    (1, 1589161),
    (2, 2781313),
    (2, 2894131),
    (3, 3471123),
    (3, 3857841),
    (4, 4938732),
    (4, 4538713),
    (5, 5982412),
    (5, 5001378),
    (6, 6982412),
    (6, 6001378);

INSERT INTO cell(rack_id, unique_cell_number) VALUES
    (1, 1189313),
    (1, 1289431),
    (2, 2178941),

```

```
(2, 2278914),
(3, 3178913),
(3, 3289234),
(4, 4138732),
(5, 5182412),
(6, 6182412),
(7, 7178123),
(8, 8189723),
(9, 9136781),
(10, 8901331),
(11, 1678921),
(12, 8913124);
```

```
INSERT INTO document(theme_name, document_title, inventory_number,
cell_id) VALUES
('учет инвентаря', 'Учтенный инвентарь за 2023', '01000000012023',
1),
('документация', 'Правила инвентаризации', '01000100012023', 4),
('отчеты экономистов', 'План экономического развития на 2024',
'02000020242023', 5),
('налоговая', 'Результаты налоговой инспекции за 2022',
'02000011112022', 7),
('документация', 'Комплектная документация на станки от Bosh',
'03000019982019', 8),
('история предприятия', 'История предприятия с момента октябрьской
революции', '07108104522010', 9),
('охрана труда', 'Политика организации по охране труда',
'08000035642010', 10);
```

```
INSERT INTO instance(title, inventory_number, status, document_id) VALUES
('Учтенные станки', '0000000010001', 'есть', 1),
('Правила инвентаризации экз.1', '000100010001', 'есть', 2),
('Правила инвентаризации экз.2', '000100010002', 'есть', 2),
('Новый экономический план на предприятии', '000020240001', 'есть',
3),
('Документация на станок BOSH №313ADC31', '000019980001', 'есть', 5);
```

```
INSERT INTO department(department_name, department_number) VALUES
('материально-технического снабжения', '013-00-06'),
('планово-экономический', '916-14-61'),
('конструкторский', '343-74-62'),
('управления персоналом', '949-25-48'),
('военизированная охрана', '045-22-26');
```

```
INSERT INTO abonent(phone_number, surname, name, middle_name,
department_id) VALUES
('7(416)127-21-37', 'Осипова', 'Мария', 'Дмитриевна', 1),
('7(589)183-65-03', 'Кириллов', 'Артём', 'Робертович', 1),
('7(842)993-16-33', 'Богданова', 'Эмилия', 'Александровна', 2),
('7(712)011-66-67', 'Кузнецов', 'Фёдор', 'Никитич', 3),
('7(404)450-48-49', 'Буров', 'Илья', 'Лукич', 5);
```

```
INSERT INTO issues(instance_id, abonent_id, date_of_issue) VALUES
(1, 4, '2023-09-01'),
(5, 4, '2023-10-03'),
(2, 1, '2023-10-12'),
(3, 2, '2023-10-15');
```

```

(4, 5, '2023-10-20');

INSERT INTO issues_archive(document_id, date_of_issue, surname, name,
middle_name) VALUES
    (6, '2010-01-01', 'Робертов', 'Артём', 'Данилович'),
    (7, '2013-10-18', 'Соколова', 'Софья', 'Тимуровна'),
    (5, '2020-11-11', 'Мухин', 'Антон', 'Михайлович'),
    (5, '2022-07-15', 'Богданова', 'Эмилия', 'Александровна'),
    (5, '2016-09-24', 'Буров', 'Илья', 'Лукич');

-- Название наиболее востребованного документа?
WITH all_issues AS (
    SELECT document_id, date_of_issue FROM issues
        INNER JOIN instance ON instance.instance_id =
issues.instance_id
    UNION ALL
    SELECT document_id, date_of_issue FROM issues_archive
)
SELECT document_title, COUNT(*) AS Количество_выдач FROM all_issues
    INNER JOIN document ON all_issues.document_id = document.document_id
    GROUP BY document.document_id
    HAVING COUNT(*) = (SELECT COUNT(*) FROM all_issues GROUP BY
document_id ORDER BY 1 DESC LIMIT 1);

-- Общее количество документов на заданную тему?
SELECT COUNT(*) AS Количество FROM document
    WHERE theme_name = 'документация';

-- Тема документа по заданному названию?
SELECT theme_name AS Тема FROM document
    WHERE document_title = 'План экономического развития на 2024';

-- Название документа, который имеется в архиве в максимальном количестве
экземпляров?
SELECT document.document_id, document_title AS Название FROM document
    INNER JOIN instance ON document.document_id = instance.document_id
    GROUP BY document.document_id
    HAVING COUNT(*) = (SELECT COUNT(*) FROM instance
        WHERE status <> 'утерян'
        GROUP BY document_id
        ORDER BY 1 DESC
        LIMIT 1);

-- Фамилия, имя и отчество абонента, который брал указанный документ
последним?
SELECT Фамилия, Имя, Отчество FROM (
    SELECT surname AS Фамилия, name AS Имя, middle_name AS Отчество,
date_of_issue, document_id FROM issues_archive
    UNION ALL
    SELECT surname AS Фамилия, name AS Имя, middle_name AS Отчество,
date_of_issue, document_id FROM issues
        INNER JOIN abonent ON abonent.abonent_id = issues.abonent_id
        INNER JOIN instance ON instance.instance_id = issues.instance_id
) AS takers_info
WHERE document_id = (SELECT document_id FROM document WHERE
document_title='Политика организации по охране труда')
ORDER BY date_of_issue DESC

```

```

LIMIT 1;

-- Есть ли в архиве пустые стеллажи, полки, ячейки, и в каком количестве?
WITH free_cells AS (
SELECT cell.cell_id, rack.rack_id, shelf.shelf_id, CASE WHEN document_id
IS NULL THEN 1 ELSE 0 END AS cell_empty FROM cell
    INNER JOIN rack ON rack.rack_id = cell.rack_id
    INNER JOIN shelf ON shelf.shelf_id = rack.shelf_id
    LEFT JOIN document ON document.cell_id = cell.cell_id
),
free_racks AS (
    SELECT rack_id FROM free_cells
    GROUP BY rack_id
    HAVING COUNT(*) = SUM(cell_empty)
),
free_shelves AS (
    SELECT shelf_id FROM free_cells
    GROUP BY shelf_id
    HAVING COUNT(*) = SUM(cell_empty)
)
SELECT 'Ячеек' AS Название, SUM(cell_empty) AS Количество FROM free_cells
UNION ALL
SELECT 'Полки' AS Название, COUNT(rack_id) AS Количество FROM free_racks
UNION ALL
SELECT 'Стеллажей' AS Название, COUNT(shelf_id) AS Количество FROM
free_shelves;

-- Список документов, не востребованных в течение более, чем 5 лет?
WITH all_issues AS (
    SELECT document_id, date_of_issue FROM issues
        INNER JOIN instance ON instance.instance_id =
issues.instance_id
    UNION ALL
    SELECT document_id, date_of_issue FROM issues_archive
)
SELECT document.document_id, document_title, theme_name,
MAX(date_of_issue) AS Дата_последней_выдачи FROM document
    LEFT JOIN all_issues ON document.document_id = all_issues.document_id
    GROUP BY document.document_id
    HAVING MAX(date_of_issue) IS NULL OR age(MAX(date_of_issue)) >
interval '5 years';

```