|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»

**Клиент-серверные системы управления банком данных**

**Практическая работа № 6**

**Подключение и выгрузка сторонних неструктурированных данных**

**ОТЧЕТ**

Выполнил студент группы:

БСБО-07-20

Любовский С.В.

**Москва 2022г.**

Задание на практику:

1. На основе практических работ 2-5 разработать механизм отчета по заданиям, который выгружается в формате .json и может храниться и обрабатываться в нереляционных базах данных (MongoDB, CouchDB, Cassandra, CockroachDB **- выбрать одну базу данных из перечисленных**).
2. Выполнить 5 запросов к отчету через выбранную базу данных.

Отчет оформить в формате doc (docx) или pdf и выслать на проверку

**Выполнение задания**

1. Создадим функцию для получения отчетов в формате json.

Листинг функции reports\_to\_json:

CREATE OR REPLACE FUNCTION "shop"."reports\_to\_json"("users" varchar(100)[], "start\_date" date, "end\_date" date)

RETURNS JSONB

LANGUAGE plpgsql

AS

$$

DECLARE

    u varchar(100);

    reports JSONB := '[]'::JSONB;

BEGIN

    foreach u in array "users"

    loop

         reports = reports || (SELECT ROW\_TO\_JSON(t) :: JSONB FROM (SELECT \* FROM shop.create\_report(u, "start\_date", "end\_date")) t);

    end loop;

    RETURN reports;

END;

$$

2. Создадим отчеты с использованием функции reports\_to\_json.

Листинг консольной команды для создания файла reports.json с отчетами в формате json:

psql -h localhost -d postgres -U postgres -c "COPY (select shop.reports\_to\_json(array['default\_worker1', 'default\_worker2'], current\_date - 15, current\_date)) TO STDOUT" > ./reports.json

3. В качестве нереляционной базы выбрана **MongoDB**. Подключимся к серверу **MongoDB** в утилите mongo-shell:

mongo -u {ПОЛЬЗОВАТЕЛЬ} -p {ПАРОЛЬ}

4. Создадим новую БД:

use shop;

5. Создадим коллекцию reports и добавим в нее наши отчеты:

db.reports.save({ДАННЫЕ\_ИЗ\_ФАЙЛА\_reports.json})

6. Произведем поиск.

1. Запрос с условием по работнику:

db.reports.find({"user": "default\_worker1"})

1. Запрос для сортировки отчетов по дате начала периода:

db.reports.aggregate([{$addFields:{date\_start:{$toDate: "$date\_start"}}},{ $sort: { date\_start: -1 }}])

1. Запрос для получения отчетов в которых нет незавершенных заданий:

db.reports.find({ "not\_completed\_task\_count": 0, "not\_completed\_out\_of\_date\_task\_count": 0 })

1. Запрос с ограничением по дате начала отчета:

db.reports.aggregate( [ { $addFields: { date\_start: { $toDate: "$date\_start"} } }, { $match: {"date\_start": { $gt: new Date('2022-10-09') } } } ])

1. Запрос с удалением отчетов дата которых меньше заданной:

db.reports.remove({$expr: {$lt: [ { $dateFromString: {dateString: "$date\_start"} }, new Date('2022-10-09')]}})