

## РАЗДЕЛ 4 ТЕХНИЧЕСКИЕ НАУКИ

### 4.1 Информационные системы и автоматизация производства

УДК 004.652

#### РАЗРАБОТКА NOSQL БАЗ ДАННЫХ ТИПА «КЛЮЧ-ЗНАЧЕНИЕ» В ПРЕДМЕТНОЙ ОБЛАСТИ

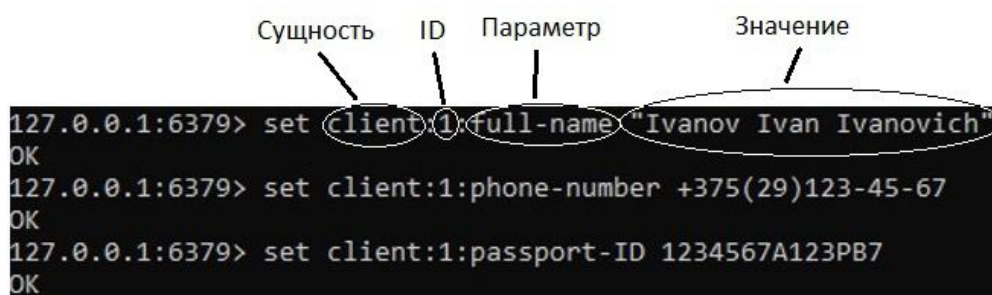
Агапитов М.Н., студ., Черненко Д.В., ст. преп., Куксевич В.Ф., ст. преп.  
*Витебский государственный технологический университет,  
г. Витебск, Республика Беларусь*

СУБД NoSQL (No only SQL) – это реализация масштабируемого хранилища информации с гибкой моделью данных, отличающаяся от классических реляционных СУБД.

Из четырех основных типов NoSQL СУБД выберем к использованию хранилище типа «ключ-значение». Базы данных типа «ключ-значение», как правило, используют хеш-таблицу, в которой находится уникальный ключ и указатель на конкретный объект данных. Такие хранилища удобны в плане доступности и устойчивости к разделению, но проигрывают в согласованности данных [1].

В качестве основы предлагаемой разработки будем использовать СУБД NoSQL-типа Redis, хорошо зарекомендовавшую себя среди СУБД типа «ключ-значение», для ведения БД страховой компании.

Вначале создадим учетную запись клиента компании, в которой будут присутствовать следующие данные: ФИО, номер телефона, идентификационный номер. При помощи команды `set` будем создавать сущности (параметры) и заполнять их значениями (рис. 1). Для большего понимания будем использовать единую схему для ключей «Сущность : ID : Параметр».



```
127.0.0.1:6379> set client:1:full-name "Ivanov Ivan Ivanovich"
OK
127.0.0.1:6379> set client:1:phone-number +375(29)123-45-67
OK
127.0.0.1:6379> set client:1:passport-ID 1234567A123PB7
OK
```

The image shows a terminal window with three Redis `set` commands. Labels with arrows point to specific parts of the keys: 'Сущность' points to 'client', 'ID' points to '1', 'Параметр' points to 'full-name', and 'Значение' points to the value 'Ivanov Ivan Ivanovich'.

Рисунок 1 – Учетная запись клиента

Аналогично создаем учетные записи агента (с параметрами: ФИО, номер телефона, дата приема на работу) и договора (с параметрами: ID клиента, ID агента, дата заключения, страховая сумма, вид страхования имущества).

В результате практического использования разработки можно сделать вывод, что подобные базы данных, рассчитанные на приложения, выполняющие простые операции поиска на основе значения ключа или диапазона ключей, легко масштабируется, позволяя удобно распределить данные среди нескольких узлов на разных компьютерах. Однако они не очень подходят при необходимости запрашивать данные из нескольких таблиц и неудобны в сценариях, где выполняются запросы или фильтрация по значению, а не только по ключам.

#### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Разбираемся в типах NoSQL СУБД [Электронный ресурс]. – Режим доступа: <https://tproger.ru/translations/types-of-nosql-db>. – Дата доступа: 12.05.2022.

УДК 681.5:621.865.5

## РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ РОБОТА-ТЕЛЕЖКИ

**Бувевич Т.В., к.т.н., доц., Туманов В.С., асс., Садыков А.С., студ.**

*Витебский государственный технологический университет,  
г. Витебск, Республика Беларусь*

Функция робота-тележки заключается в транспортировании контейнеров заданного типа из зоны приёма в зону сбора. Также поставлены задачи распознавания типа контейнера, его захвата и подъема. Количество контейнеров заранее неизвестно. Таким образом, технические требования к системе управления робота-тележки:

- вывод изображения с камеры в режиме online без задержек;
- возможность переключения между ручным и автоматическим режимами управления;
- возможность считывания QR, а также BAR кодов с помощью камеры;
- возможность считывания, распознавания цветов и оттенков предметов, попавших в обзор камеры;
- вывод полной информации о нахождении робота в пространстве, а также вывод данных со всех его датчиков;
- передвижение по линии посредством систем технического зрения;
- вывод и отслеживание уровня заряда аккумулятора.

Для написания программного обеспечения системы управления робота-тележки использовались языки программирования Java, JavaScript, C. При создании графического интерфейса для обеспечения эффективной и быстрой работы с роботом использовано веб-приложение. Веб-приложение написано на языке Java с применением технологии сервлетов. В качестве шаблонизатора использовался JSP. Для написания страниц GUI использовался язык разметки HTML, каскадные таблицы стилей CSS, предоставленные фреймворком для разработки веб-страниц Bootstrap.

Написанный на языке программирования Java веб-сервер был развёрнут на одноплатном компьютере Raspberry Pi 3B – открытой плате, на которой расположены процессор с оперативной памятью, разъёмы для подключения HDMI и MIPI DSI, два порта