

особенностям;

– внедрить систему мониторинга срока носки спецодежды с использованием технологий RFID или штрих-кодов для автоматического определения необходимости замены.

Таким образом, применение пространственных СУБД в системах автоматизации выдачи спецодежды представляет собой перспективное направление для повышения эффективности управления данным процессом на промышленных предприятиях. Результаты данной разработки могут быть полезны для компаний, занимающихся созданием систем управления трудовыми ресурсами, а также для предприятий, стремящихся к оптимизации своих производственных процессов. Предполагается, что разработанная модель системы и созданный прототип позволят повысить эффективность и безопасность труда работников, а также оптимизировать расходы на обеспечение персонала спецодеждой.

УДК 004.65

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ЗАДАЧАМИ С ИСПОЛЬЗОВАНИЕМ ХРАНИЛИЩА ДАННЫХ WCS

*Скребло Е.И., студ., Черненко Д.В., ст. преп., Куксевич В.Ф., ст. преп.,
Соколова А.С., ст. преп.*

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

Реферат. В статье рассмотрены вопросы оптимального управления задачами выполнения профессиональных функций сотрудников компании, представлены преимущества колоночных баз данных WCS и их основные отличия от реляционных баз данных, приведены этапы проектирования базы данных «Управление задачами».

Ключевые слова: управление задачами, автоматизированная система управления, базы данных, СУБД WCS.

Ключевую роль в реалиях современного бизнеса играет оптимальное управление задачами. Для выполнения своих профессиональных функций сотрудники компании получают различные задачи. В сфере информационных технологий это могут быть: разработка, тестирование, анализ и т. д. При этом каждая задача имеет определенный срок выполнения, по истечении которого она предполагается быть успешно реализованной.

В процессе реализации автоматизированной системы управления задачами сотрудников компании возможно использование Apache Cassandra, открытой и распределенной базы данных NoSQL, реализующей модель Wide Column Stores (WCS).

Преимущества колоночных баз данных WCS в скорости выполнения запросов, гибкости модели данных и масштабируемости. WCS относится к NoSQL базам данных, хранение данных в которых производится с помощью гибких столбцов, распределяемых по разным серверам, с использованием многомерного сопоставления для ссылки на данные.

Основное отличие СУБД WCS от систем управления реляционными базами данных состоит в том, что последние хранят данные в таблицах со строками, охватывающими несколько столбцов. И если для какой-либо строки необходим дополнительный столбец, то он добавляется к таблице со значениями NULL. В случае запроса СУБД в этой таблице значения, которое не проиндексировано, поиск такого значения будет достаточно медленным. СУБД WCS, как и реляционных баз данных, имеют концепцию строк, но чтение (запись) поля данных состоит из чтения (записи) отдельных столбцов. При этом столбец записывается только в случае присутствия в нем элемента данных, а запрос значения оптимизирован так же, как запрос индекса в СУБД.

Применение Apache Cassandra позволяет эффективно управлять большими объемами данных и обеспечивает высокую производительность автоматизированной системы управления задачами. Для удобства разработки и тестирования был использован редактор кода Visual Studio Code с расширением Cassandra Workbench, что значительно упростило процесс работы с базой данных, обеспечило гибкость и эффективность реализации проекта.

Для эффективного управления задачами компании была спроектирована база данных

«Управление задачами». Информация из этой базы данных может использоваться для получения оперативных данных о статусе задач у сотрудников, формирования списка сотрудников, нуждающихся в выполнении задач, планирования распределения задач.

В базе данных будет храниться следующая информация:

- о задачах: тип задачи и сроки ее выполнения;
- о проектах: название проекта и список сотрудников, работающих над ним;
- о сотрудниках: Ф.И.О. сотрудника и его контактные данные.

При проектировании базы данных предполагается, что каждый сотрудник может работать только над одним проектом. При этом каждый проект может включать в себя несколько задач, однако каждая задача относится только к одному проекту и выполняется одним сотрудником.

Использование системы с WCS позволит улучшить процесс управления задачами в компании, повысить эффективность работы сотрудников и обеспечить своевременное выполнение всех задач.

Перед созданием таблиц и заполнением их данными в Apache Cassandra создается пространство ключей аналогично реляционным базам данных. Код для создания пространства ключей представлен на рисунке 1.

```
CREATE KEYSPACE my_keyspace_for_wcs WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 3};
```

Рисунок 1 – Код для создания пространства ключей

Далее переходим к созданию таблиц. Создаем три таблицы (Userss, Projects и Tasks) и вставляем данные в них с помощью оператора INSERT INTO. Это позволит заполнить таблицы данными и начать работу с системой управления задачами. Для проверки данных и обеспечения корректной работы системы управления задачами осуществим просмотр содержимого таблиц, используя оператор SELECT * FROM (рис. 2).

```
-- Вывод данных для пользователей
SELECT * FROM Userss;
```

	userss_id	additional_Info	email	registration_date	username
1	user_1	Стажер	ivanov@example.com	2024-03-27	Иванов
2	user_2		petrov@example.com	2024-03-20	Петров

```
-- Вывод данных для проектов
SELECT * FROM Projects;
```

	project_id	description_p	end_date	start_date	title	userss_id
1	project_2	Разработка мобильного приложения для iOS и Android	2024-04-08	2024-04-01	Мобильное приложение	user_2
2	project_1	Создание корпоративного сайта для компании	2024-04-06	2024-03-28	Разработка сайта	user_1

```
-- Вывод данных для задач
SELECT * FROM Tasks;
```

	task_id	completion_date	creation_date	description_t	project_id	status	title
1	task_1	2024-04-02	2024-03-29	Создание макета дизайна сайта	project_1	Завершено	Дизайн сайта
2	task_2	2024-04-06	2024-04-02	Создание пользовательского интерфейса для мобильного приложения	project_2	В процессе	Разработка UI

Рисунок 2 – Просмотр содержимого таблиц

Таблица Userss предназначена для хранения информации о пользователях. Она содержит следующие поля: userss_id (уникальный идентификатор пользователя), username (имя пользователя), email (электронная почта пользователя) и registration_date (дата регистрации пользователя).

Таблица Projects содержит информацию о проектах. Она включает в себя поля: project_id (уникальный идентификатор проекта), userss_id (идентификатор пользователя, работающего над проектом), title (название проекта), description_p (описание проекта), start_date (дата начала

проекта) и end_date (дата окончания проекта).

Таблица Tasks предназначена для хранения информации о задачах. Она содержит следующие поля: task_id (уникальный идентификатор задачи), project_id (идентификатор проекта, к которому относится задача), title (название задачи), description_t (описание задачи), status (статус задачи), creation_date (дата создания задачи) и completion_date (дата завершения задачи).

Одной из особенностей Apache Cassandra является возможность в каждой строке иметь различное количество столбцов. Данное преимущество позволяет по мере необходимости добавлять дополнительные столбцы в определенные строки. Пример кода, демонстрирующего такую возможность, приведен на рисунке 3.

```
USE my_keyspace_for_wcs;
ALTER TABLE Userss ADD additional_info text;
-- Добавление дополнительного столбца в определенную строку
UPDATE Userss SET additional_info = 'Стажер' WHERE userss_id = 'user_1';
```

Рисунок 3 – Пример кода добавления столбцов

Оператор ALTER TABLE изменяет структуру существующей таблицы, добавляя новый столбец additional_info в таблицу Userss. Оператор UPDATE изменяет данные в существующей строке таблицы, устанавливая значение Стажер для столбца additional_info в строке, где userss_id равен user_1.

Таким образом, с помощью Apache Cassandra и Visual Studio Code с расширением Cassandra Workbench, была создана эффективная и масштабируемая система управления задачами.

В современных реалиях, когда объемы данных растут ежедневно, использование модели WCS, обладающей гибкостью и масштабируемостью, для эффективной работы с большими данными может обеспечить высокую производительность, необходимую для современных приложений. Это является важным шагом в цифровой трансформации бизнес-процессов и повышении производительности труда.

Список использованных источников

1. Виды баз данных. Большой обзор типов СУБД [Электронный ресурс]. – Режим доступа: <https://www.pvsm.ru/grafovy-e-bazy-danny-h/386823>. – Дата доступа: 15.04.2024.
2. Overview | Apache Cassandra Documentation [Электронный ресурс]. – Режим доступа: <https://cassandra.apache.org/doc/latest/cassandra/architecture/overview.html>. – Дата доступа: 16.04.2024.

УДК 62-236.58

РАЗРАБОТКА ЛАБОРАТОРНОГО ОБРАЗЦА МАНИПУЛЯТОРА

Трусов А.С., студ., Белов А.А., к.т.н., доц.

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

Реферат. В статье рассмотрены вопросы проектирования лабораторного образца манипулятора, представлена кинематическая схема и общий вид манипулятора.

Ключевые слова: роботизированная платформа Robotino 3, манипулятор, схват, роботизированные системы.

Роботизированные системы стали неотъемлемой частью современной индустрии и автоматизации процессов. Они способны выполнять разнообразные задачи, от монтажа деталей на производстве до доставки товаров в складах. В рамках этой статьи будет рассмотрена разработка манипулятора для роботизированной платформы Robotino 3.

Robotino 3 представляет собой универсальную мобильную роботизированную платформу, предназначенную для выполнения различных задач в сфере автоматизации и робототехники.