

### Источники литературы

1. Информационные технологии в бизнесе. [Электронный ресурс] – Режим доступа: <http://economyz.ru/informatsionnye-tekhnologii-v-biznese>– Дата доступа: 26.11.2018
2. Роль IT в бизнес процессах [Электронный ресурс] – Режим доступа: <https://www.stekspb.ru/blog/it/rol-it-v-biznes-processah>– Дата доступа: 26.11.2018
3. П. Прохоров, В.С. Информационные технологии управления: учебное пособие / В.С. Прохоров.— Москва : Издательство Флинта., 2008.— 87 с.
4. Буян, Р.А. Роль информационных технологий в управлении структурой бизнес-процессов: учебное пособие / Р.А. Буян.— Москва : Издательство креативная экономика., 2009.—Том 10. – № 5. – 64-70 с.

*Pilipchuk Hanna*

*Belarus state economic university*

### **The role of information technology in business**

Annotation: This article focuses on defining the role of information technology in a business system. This article reveals the reasons for the popularization of information systems in all areas of the production of goods and services, identifies the main blocks of business systems, and describes the role of information technologies in each of them: management, production, marketing. The role of information technology is also shown as an instrument uniting these three blocks and connecting them into a single business system.

Key words: information technology, management, production, marketing, business, organization, market, competition, system, structure.

УДК 00.004.655.3

*Покрушинская Алина Олеговна, Козловская николь Евгеньевна*  
*Белорусский государственный экономический университет*  
*a.pokrushinskaya@yandex.ru nikol.kozlovskaya.99@mail.ru*

### **Язык NoSQL**

Реляционные базы данных обычно не предназначены для того, чтобы справляться с проблемами масштабируемости и гибкости, которые наблюдаются в современных приложениях, и не созданы, чтобы использовать преимущества дешевой системы хранения и обработки, которая доступна в настоящее время через облако. Поэтому целью исследования является проанализировать проблемы, возникающие при работе с большими объёмами данных, а также исследовать и выявить достоинства системы NoSQL.

Поставщики реляционных баз данных разработали два основных технических подхода для устранения этих недостатков:

- ручная настройка;
- распределённый кэш.

С помощью ручной настройки все таблицы разбиваются на более мелкие физические объекты и распространяются на несколько серверов. Поскольку база данных не обеспечивает эту способность изначально, команды разработчиков берут на себя работу по



развертыванию нескольких реляционных баз данных на нескольких машинах. Данные сохраняются в каждом экземпляре базы данных автономно. Код приложения разработан для распространения данных, распространения запросов и агрегирования результатов данных во всех экземплярах базы данных. Дополнительный код должен быть разработан для обработки сбоев ресурсов, для выполнения объединений в разных базах данных, для поддержания баланса данных, репликации и других требований. Кроме того, многие преимущества реляционной базы данных, такие, как целостность транзакций, устранены при использовании ручного настраивания [1].

Ряд продуктов обеспечивают уровень кэширования для систем баз данных. Эти системы могут существенно улучшить производительность чтения, но они не улучшают производительность записи, вдобавок к этому усложняют развертывание системы. Если в приложении преобладает чтение, тогда, следует учитывать распределенный кэш. Однако если в приложении преобладают записи или сочетание чтения и записи равномерное, тогда распределенный кэш может не улучшить общий опыт вашего конца пользователя. Базы данных NoSQL появились в ответ на эти вызовы и в ответ на новые возможности, предоставляемые недорогими аппаратными средствами и облачными средами развертывания, и изначально поддерживают современную среду развертывания приложений, что уменьшает необходимость в том, чтобы разработчики поддерживали отдельные уровни кэширования.

Стоит подчеркнуть, что термин «NoSQL» имеет абсолютно стихийное происхождение и не имеет общепризнанного определения или научного учреждения за спиной. Это название скорее характеризует вектор развития ИТ в сторону от реляционных баз данных. Расшифровывается как «Not Only SQL», хотя есть сторонники и прямого определения No SQL [1].

Общих характеристик для всех NoSQL немного, так как под данным лэйблом сейчас скрывается множество разнородных систем. Основными являются динамические системы, интегрированное кэширование и масштабирование.

Реляционные базы данных требуют определение схем для добавления данных. Например, при необходимости хранить данные о клиентах, телефонные номера, имя и фамилию, адрес, город и страну \_ база данных SQL должна знать это заранее [2].

Это плохо подходит для гибких подходов к разработке, поскольку каждый раз, когда заполняются новые функции, часто необходимо изменить схему базы данных. Если база данных большая, это очень медленный процесс, который связан с существенным временем простоя. Если часто изменять данные, хранящиеся в приложении, это время простоя также может быть частым. Не разработан способ использовать реляционную базу данных для эффективного устранения данных, которые полностью не структурированы или неизвестны заранее.

Базы данных NoSQL построены так, чтобы вставлять данные без predefined схемы. Это упрощает внесение значительных изменений приложений в режиме реального времени, не беспокоясь о прерываниях службы, что означает, что разработка выполняется быстрее, интеграция кода более надежна и требуется меньше времени администратора базы данных.

Из-за структуры реляционные базы данных обычно масштабируются вертикально - на одном сервере должна размещаться вся база данных, чтобы обеспечить надежность и постоянную доступность данных. Это быстро становится дорогостоящим, ограничивает масштаб и создает относительно небольшое количество точек отказа для инфраструктуры базы данных. Решение заключается в масштабировании по горизонтали, добавляя серверы вместо концентрации большей емкости на одном сервере. Облачные вычисления делают это значительно проще: провайдеры предоставляют практически неограниченные возможности по требованию и заботятся обо всех необходимых задачах администрирования базы данных. Разработчикам больше не нужно создавать сложные дорогостоящие платформы для поддержки приложений. Таким образом, можно полностью сосредоточиться на написании кода приложения. Кроме того, группа товарных серверов может предоставлять одни и те же возможности обработки и хранения как один сервер высокого класса за небольшую часть цены. Обработка базы данных на многих экземплярах сервера может быть достигнута с базами данных SQL, но обычно она выполняется через SAN и другие сложные механизмы для создания аппаратного обеспечения как единого сервера.

С другой стороны, базы данных NoSQL обычно поддерживают автоматическое очертание, что означает, что они изначально и автоматически распространяют данные на произвольном количестве

серверов, не требуя от приложения даже знать о составе пула серверов. Загрузка данных и запросов автоматически распределяется между серверами, а когда сервер опускается, его можно быстро и прозрачно заменить без нарушения работы приложения.

В отличие от реляционных баз данных, базы данных NoSQL обычно не требуют отдельных приложений или дорогостоящих надстроек для реализации репликации. Большинство баз данных NoSQL также поддерживают автоматическую репликацию для обеспечения доступности в случае сбоев или запланированных событий обслуживания. Более сложные базы данных NoSQL полностью самовосстанавливаются, предлагая автоматическое восстановление после сбоев и восстановление, а также возможность распространять базу данных в разных географических регионах, чтобы противостоять региональным сбоям и обеспечивать локализацию данных. Среда хранения практически виртуализована с точки зрения разработчика. Наконец, многие технологии баз данных NoSQL обладают отличными интегрированными возможностями кэширования, позволяя максимально использовать часто используемые данные в системной памяти. Это устраняет необходимость в отдельном кэшировании, который должен поддерживаться.

Существует ещё несколько существенных преимуществ NoSQL. По сравнению с реляционными базами данных базы данных NoSQL более масштабируемые и обеспечивают превосходную производительность, а их модель данных затрагивает несколько проблем, которые реляционная модель способна решить.

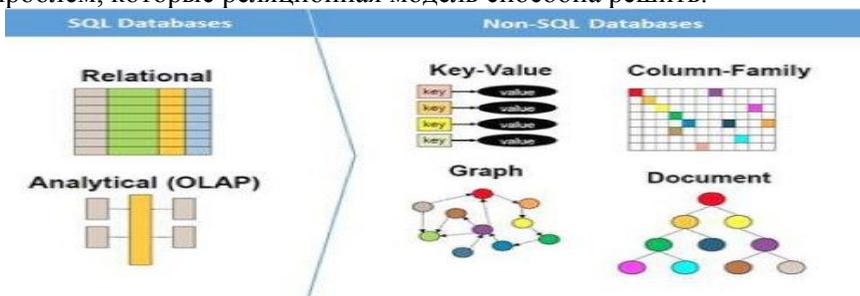


Рисунок 1 – SQL и NoSQL базы данных

Примечание – Источник [3]

Таким образом, в современном мире, где объёмы располагаемой информации становятся всё более большими, такой язык, как NoSQL необходим для обработки и анализа данных.



### Источники литературы

1. What is NoSQL? // MongoDB – Open source document database. [Электронный ресурс]. Режим доступа: <https://www.mongodb.com/nosql-inline>. – Дата доступа: 01.12.2018.

2. Клеменков П.А. Большие данные: современные подходы к хранению и обработке. / П.А. Клеменков, С.Д. Кузнецов // Труды Института системного программирования РАН. – 2012. – № 23. – С. 143–158.

3. 7 Steps to Understanding NoSQL Databases.// MongoDB – KD nuggets. [Электронный ресурс]. Режим доступа: <https://www.kdnuggets.com/2016/07/seven-steps-understanding-nosql-databases.html>. – Дата доступа: 01.12.2018.

*Pokrushinskaya Alina, Kozlovskaya Nikol*

*Belarus state economic university*

### NoSQL language

Annotation. Large amounts of data have posed quite complex tasks for traditional database systems. This article analyzes the methods for solving this problem, the limitations that do not allow to do this effectively, and you can also view an overview of the data from the system: NoSQL.

Key words: Relational databases, caching, NoSQL, replication.

УДК 331.556.4

*Полецук Александра Викторовна, Лазакович Анна Владимировна*

*Белорусский государственный экономический университет*

*polesukaleksandra99@gmail.com, anna27659@gmail.com*

### Эмиграция трудовых ресурсов: «утечка мозгов» из Республики Беларусь

Многие страны зачастую сталкиваются с такой проблемой, как миграция трудовых ресурсов за рубеж. Естественная миграция существовала всегда, но в настоящее время в связи с поиском «лучшей жизни и работы» появляется такое понятие как «утечка мозгов». Такая проблема актуальна и для Республики Беларусь.

Целью данного исследования является выявление основных причин «утечки мозгов» и предложение возможных способов удержания квалифицированных кадров.

«Утечка мозгов» - это миграция высококвалифицированных специалистов за рубеж на постоянную работу с целью улучшения своего материального положения, поиска возможностей самореализации и вследствие других социально-экономических и политических причин [1].

Это понятие предполагает получение определенного экономического эффекта принимающей страной. Например, белорусские специалисты, эмигрирующие за рубеж, осваивают новые технологии, повышают квалификацию, получают определенный опыт в организации трудового процесса, совершенствуют знания

