

Разработка мероприятий в рамках обеспечения конкурентоспособности предприятий должна осуществляться на основе следующих принципов:

- принцип системности;
- принцип функциональной направленности;
- принцип учета специфики деятельности;
- принцип информационного обеспечения;
- принцип использования современных технологий и инструментов;
- принцип комплексной оценки факторов.

Таким образом, управление конкурентоспособностью предприятия есть процесс, а система управления конкурентоспособностью – механизм, который обеспечивает этот процесс.

Конкурентная среда является многофакторной и многоуровневой категорией, используемой как для текущего, так и для стратегического планирования развития потребительского рынка. Следовательно, система обеспечения конкурентоспособности сетевых торговых предприятий состоит из набора механизмов образующих элементов, которые являются сложными объектами, определенным образом взаимосвязанными между собой. Степень агрегированности этих элементов предопределяет устойчивость и эффективность функционирования всей системы. Интенсивные темпы развития сетевого формата торговли обуславливают формирование конкурентной среды в данном сегменте потребительского рынка. В такой ситуации необходим механизм выявления взаимосвязей инструментов и методов, обеспечивающих конкурентоспособность объектов в условиях обостряющейся конкуренции.

Обеспечение конкурентоспособности является центральной задачей, стоящей перед предприятием, поэтому в современных условиях сетевые розничные торговые предприятия нуждаются в адекватных инструментах формирования и обеспечения конкурентоспособности.

*Е.А. Колотаев*

Научный руководитель *Е.П. Панфилова*

*Филиал БГЭУ (Бобруйск)*

## **ПРИМЕНЕНИЕ НЕРЕЛЯЦИОННЫХ БАЗ ДАННЫХ НА ПРИМЕРЕ MONGODB**

В последнее время появилось новое семейство баз данных. Это, так называемые, нереляционные базы данных, NoSQL. Сторонниками концепции NoSQL подчеркивается, что она не является полным отрицанием языка SQL и реляционной модели, проект исходит из того, что SQL – это важный и весьма полезный инструмент, но при этом он не может считаться универсальным. Одной из проблем, которую указывают для классических реляционных БД, являются проблемы при работе с данными очень большого объема и в проектах с высокой нагрузкой. Основная цель подхода – расширить возможности БД там, где SQL недостаточно гибок, и не вытеснять его там, где он справляется со

своими задачами. В качестве одного из методологических обоснований подхода NoSQL используется эвристический принцип, известный как теорема CAP, утверждающий, что в распределенной системе невозможно одновременно обеспечить согласованность данных, доступность (англ. availability, в смысле наличия отклика по любому запросу) и устойчивость к расщеплению распределенной системы на изолированные части. Таким образом, при необходимости достижения высокой доступности и устойчивости к разделению, предполагается не фокусироваться на средствах обеспечения согласованности данных, обеспечиваемых традиционными SQL-ориентированными СУБД с транзакционными механизмами на принципах ACID.

Одним из представителей этого семейства является БД MongoDB.

MongoDB – документо-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++ и распространяется в рамках лицензии Creative Commons. MongoDB, по мнению разработчиков, должна заполнить разрыв между простыми хранилищами данных типа «ключ-значение» (быстрыми и легко масштабируемыми) и большими РСУБД (со структурными схемами и мощными запросами).

Что дает MongoDB:

1. *Гибкость.* MongoDB хранит данные в документах формата JSON (которые сериализованы в BSON). JSON дает богатые возможности описания модели данных, которая плавно ложится на модель описания в различных языках программирования, и поскольку модель не требует описания схемы, это упрощает развитие модели данных по сравнению с системами с обязательным описанием схем, такими как СУБД.

2. *Возможности.* MongoDB позволяет использовать многие преимущества традиционных СУБД, например вторичные индексы, динамические запросы, сортировку, расширенные обновления, обновление-вставку, легкую агрегацию. Это дает широкие функциональные возможности, которые многие привыкли использовать в реляционных базах данных в сочетании с гибкостью и возможностью масштабирования.

3. *Скорость/Масштабирование.* Сохранять данные, которые связаны в документе вместе, можно значительно быстрее, чем в реляционных базах данных, где данные разделяются на несколько таблиц, которые при необходимости позже соединяются. MongoDB позволяет с легкостью масштабировать базу данных. Автоматическое распределение позволяет масштабировать кластеры линейно за счет добавления новых машин.

4. *Простота использования.* База данных может быть запущена в любой среде на вашем сервере или на виртуальной машине, а также на облачном сервисе.

Таким образом, нереляционные базы данных не призваны полностью заменить традиционные реляционные SQL решения, они применяются лишь для того, чтобы обеспечить гибкость, горизонтальную масштабируемость и высокую скорость выполнения запросов. Зачастую они используются как промежуточное звено между реляционными базами данных и приложениями.