

УДК 004.9:67/68

РАЗРАБОТКА ПОСТРЕЛЯЦИОННЫХ БАЗ ДАННЫХ ДЛЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ ЛЕГКОЙ ПРОМЫШЛЕННОСТИ

**Черненко Д.В., ст. преп., Буевич Т.В., к.т.н., доц.,
Тестова А.Д., студ., Гришаев Д.А., студ.**

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

Автоматизация технологической подготовки производства на предприятиях требует создания собственной информационной базы, включающей перечень технологических операций, номенклатуру и технические характеристики технологического оборудования, нормы времени. Для технологических процессов легкой промышленности предлагается разработать постреляционную базу данных. Необходимость создания собственных баз данных на предприятиях обусловлена частой сменой ассортимента выпускаемых изделий и материалов для их изготовления. Постреляционные базы данных предлагают динамическую структуру данных, которые могут храниться несколькими способами: ориентированно по колонкам, документо-ориентированно, в виде графов или на основе пар «ключ – значение». Преимущества постреляционной базы данных:

- возможность создавать документы, не задавая их структуру заранее;
- собственная структура каждого документа;
- собственный синтаксис у каждой базы данных;
- возможность добавлять поля прямо во время работы с данными.

MongoDB представляет собой базу данных на основе документов. В такой базе данных уникальный ключ соединяется со сложной структурой данных – документом. Документы могут содержать много разных пар «ключ – значение», пар ключей или даже вложенные документы. Хранилище данных документов управляет набором значений именованных строковых полей и данных объекта в сущности, которая называется документом. Обычно данные в этих хранилищах содержатся в виде документов JSON. Каждое значение поля может представлять собой скалярный элемент, например, число, или сложный объект, например, список или коллекция типа «родитель – потомок». Данные в полях документа можно закодировать разными способами, например, в формате XML, YAML, JSON, BSON, или хранить в виде обычного текста. Поля документов доступны системе управления хранилищем, что позволяет приложению выполнять запросы и применять фильтры, основанные на значениях этих полей.

Документ содержит все данные одной сущности. Элементы, составляющие сущность, зависят от конкретного приложения. Например, сущность может содержать сведения о поставщике, потребителе, заказе. Один документ может содержать сведения, которые в реляционной СУБД обычно распределяются по нескольким реляционным таблицам. Хранилище документов позволяет не использовать одинаковую структуру для всех документов, что обеспечивает большую гибкость. Так, приложения могут хранить в документах разные данные в соответствии с текущими требованиями предприятия.

Для упрощения разработки документов MongoDB предоставляет специальные инструменты, которые позволяют описать их структуру. Причем в процессе разработки не представится никакого труда добавить дополнительное поле в документе к любой из записей. В

рамках Node.js сферы существует такой модуль, предоставляющий нужный нам функционал – Mongoose. Данный модуль описывает схемы документов в виде Javascript объектов. Каждая запись в MongoDB автоматически дополняется полем ObjectID, служащим идентификатором поля, позволяющим связать один документ с другим. Для связи двух документов используется специальный тип данных ObjectId, содержащий ссылку на запись из другого документа. Для демонстрации работы с MongoDB используется API для осуществления операций создания, удаления, изменения и добавления записей в документы, и инструменты фреймворка Express.js для составления схемы моделей документов (Mongoose).

Представленные возможности MongoDB позволяют создать базу данных, которая может быть адаптирована для любого технологического процесса предприятия.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Онлайн-руководство по MongoDB [Электронный ресурс]. – Режим доступа: <https://metanit.com/nosql/mongodb/>. – Дата доступа 19.03.2020.