

УДК 004.652:622

ИСПОЛЬЗОВАНИЕ СУБД MONGODB В АВТОМАТИЗАЦИИ ПРОЦЕССОВ ГОРНОГО ПРОИЗВОДСТВА

Куксевич В.Ф., Соколова А.С., Черненко Д.В.

Аннотация: в статье представлены этапы внедрения информационных технологий в производственные процессы горной промышленности и тенденции их развития. Проведено сравнение реляционных и документо-ориентированных моделей СУБД, рассмотрены варианты создания и заполнения базы данных в СУБД MongoDB по отношению к процессам горного производства.

Ключевые слова: информационные технологии, автоматизация, горная промышленность, базы данных, нереляционные СУБД.

Уровень программ, используемых для автоматизации процессов горной промышленности, за последние несколько десятков лет серьезно возрос, что закономерно привело к изменению характера деятельности горных компаний и к поступательному росту производительности их производств. Первый этап внедрения новых технологий проходил в начале 1960-х годов и был обусловлен созданием простых моделей горных месторождений для оценки содержания в них пород и тоннажа этих пород. В то же время начатая автоматизация позволила компаниям быстрее осваивать инвестиции, что привело к очередному скачку производительности [1].

В 1970-е годы в горной промышленности появилось трехмерное цифровое блочное моделирование и геостатистический анализ ресурсов. Использование этих технологий дало преимущества для расчета запасов месторождений, улучшения качества и достоверности оценки ресурсов.

Третий этап охарактеризовался появлением технологий трехмерного геометрического моделирования и визуализации, давших геологам возможность рассматривать геоструктуры как пространственные объекты и обеспечивать более точный контроль создания цифровых блочных моделей. Визуализация моделей получила долговременный эффект, внесла существенные изменения в методологию их конструирования и анализа.

Конец 1980-х был отмечен компьютеризацией таких процессов, как горное проектирование, оптимизация, календарное планирование. Это вновь позволило увеличить производительность производства, однако, как и на предыдущих этапах, волна модернизации прошла свой пик и начала спадать в середине 1990-х.

В настоящее время программное обеспечение (ПО) для технологий горного моделирования и проектирования представляет собой системы, отличающиеся интерактивной графикой, высокого качества визуализацией поверхностей и моделей объектов, дружественными интерфейсами пользователя. Даже недорогие ПК могут теперь обеспечить графическое интерактивное проектирование, а более совершенные компьютеры и программные средства помогают инженерам-геологам выполнять самые сложные процессы автоматизации горного проектирования быстро и эффективно.

Тем не менее, скачки производительности на данном этапе развития горной промышленности существенно замедлились, так как большинство предприятий используют информационные технологии для улучшения отдельных технологических процессов, а не в целом производства. Много усилий тратится для автоматизации старых методов и оборудования вместо кардинального их изменения. Следующая волна технологий, которая сейчас формируется, должна обеспечить динамическое улучшение производительности процессов в горной отрасли.

Большинство современного ПО, применяемого в том числе в горном деле, использует базы данных (БД), позволяющие упорядоченно хранить данные о группе объектов производства или проектирования, обладающих одинаковым набором свойств. До последнего времени бóльшая часть БД строилась на основании реляционной модели.

Реляционная модель данных позволяет представлять информацию о предметной области с помощью взаимосвязанных таблиц [2].

В реляционных БД вся информация сводится в таблицы, строки и столбцы, называемые соответственно записями и полями. Данные таблицы получили название реляций. Записи в таблицах не повторяются, а их уникальность обеспечивается первичным ключом, содержащим набор полей, однозначно определяющих запись.

Данная модель имеет как достоинства, так и недостатки. К достоинствам можно отнести отображение информации в наиболее простой для пользователя форме; развитый математический аппарат, позволяющий достаточно лаконично описать основные операции над данными; возможность создавать языки манипулирования данными непроедурного типа; манипулирование данными на уровне выходной БД с возможностью изменения.

Недостатками реляционной модели можно считать достаточно медленный доступ к данным и трудоемкость разработки.

Как известно, любая реляционная БД имеет стандартную схему, отражающую таблицы входящие в БД и связи между ними. Совершенно новый подход к построению БД реализует документо-ориентированная система управления базами данных (СУБД) MongoDB, не требующая описания схемы связей таблиц, а значит и минимизирующая недостатки реляционной модели. Данная СУБД основана на коллекциях различных документов и относится к так называемым NoSQL СУБД. При этом количество полей, содержание и размер этих документов, могут отличаться, то есть различные сущности не должны быть идентичными по структуре [3].

К основным преимуществам MongoDB относят: понятную структуру, максимально приближенную к модели реального объекта; легкость масштабирования; использование внутренней памяти для хранения обрабатываемых в данный момент данных, что позволяет получать более быстрый доступ; хранение данных в виде JSON-документов; поддержку динамических запросов документов (document-based query); отсутствие сложных JOIN-запросов и необходимости маппинга объектов приложения в объекты БД.

Таким образом, можно сделать вывод, что MongoDB является достаточно эффективным решением при использовании Big Data в автоматизации проектирования горных работ.

Так как MongoDB написана на C++, то она легко устанавливается на самые разные платформы: Windows, Linux, MacOS, Solaris.

Одним из популярных стандартов обмена данными и их хранения является JSON (JavaScript Object Notation), эффективно описывающий сложные по структуре данные. Способ хранения данных в MongoDB похож на JSON, хотя формально JSON и не используется. Для хранения здесь применяется формат, называемый BSON (сокращение от Binary JSON).

BSON позволяет увеличить скорость работы с данными: быстрее выполняется поиск и обработка. BSON, в отличие от хранения данных в формате JSON, имеет небольшой недостаток: данные в BSON-формате занимают больше места. С другой стороны, данный недостаток многократно компенсируется скоростью работы.

В MongoDB имеются особенности, которые надо учитывать в процессе работы:

1) Каждому ключу сопоставляется определенное значение. В реляционных БД есть четкая структура с полями и, если какое-то поле не имеет значение, ему можно присвоить значение NULL. В MongoDB, если какому-то ключу не сопоставлено значение, этот ключ просто опускается в документе и не используется.

2) В реляционных СУБД встречается такое понятие, как первичный ключ, описывающее некоторый столбец, имеющий уникальные значения. В MongoDB для каждого документа имеется уникальный идентификатор,

который называется `_id`. И если явным образом не указать его значение, MongoDB сгенерирует его автоматически.

В качестве примера рассмотрим создание и заполнение базы данных в нереляционной СУБД MongoDB в применении к процессам горного производства.

Создадим БД «Скважины» и добавим в неё следующие сущности: Скважины (Wells), Описание (Description), Диаметры (Diameters), Пробы (Samples).

```
# use Well_DB
# db.createCollection("Description")
# db.createCollection("Diameters")
# db.createCollection("Samples")
# db.createCollection("Wells")
```

Добавим документы в коллекции. Мы не указываем `_id`, из-за чего они будут сгенерированы автоматически.

Коллекция Description:

```
# db.Description.insertMany([
  { Power: 1.40, Grade: "АФМ" },
  { Power: 1.20, Grade: "АФМ 15-3" },
  { Power: 1.00, Grade: "ФМ" }
])
```

Коллекция Diameters:

```
# db.Diameters.insertMany([
  { Deep: 3.00, Diameter: 3.35 },
  { Deep: 4.40, Diameter: 4.30 },
  { Deep: 5.60, Diameter: 5.25 }
])
```

Коллекция Samples:

```
# db.Samples.insertMany([
  { Number: "1923a", Fe_gen: 30.7, P2O5: 9.0,
    Description: [ObjectId("61949f8814e7b69fb36e4f3e")] },
  { Number: "1924a", Fe_gen: 22.2, P2O5: 9.0,
    Description: [ObjectId("61949f8814e7b69fb36e4f3f")] },
  { Number: "1925a", Fe_gen: 22.3, P2O5: 9.1,
    Description: [ObjectId("61949f8814e7b69fb36e4f3d")] }
])
```

Коллекция Wells:

```
# db.Wells.insertMany([
```

```
{Name: "100", Date: "2022-04-21T18:25:43-05:00", Horizon: 262,  
Diameters_id: [ObjectId("61949fa714e7b69fb36e4f40")],  
Samples_id: [ObjectId("6194a0c314e7b69fb36e4f46")]},  
{Name: "100c", Date: "2021-04-21T18:25:43-05:00", Horizon: 250,  
Diameters_id: [ObjectId("61949fa714e7b69fb36e4f40")],  
Samples_id: [ObjectId("6194a0c314e7b69fb36e4f47")]},  
{Name: "101", Date: "2022-05-21T18:25:43-05:00", Horizon: ,  
Diameters_id: [ObjectId("61949fa714e7b69fb36e4f41")],  
Samples_id: [ObjectId("6194a0c314e7b69fb36e4f48")]},  
{Name: "101c", Date: "2021-05-21T18:25:43-05:00", Horizon: 250,  
Diameters_id: [ObjectId("61949fa714e7b69fb36e4f41")],  
Samples_id: [ObjectId("6194a0c314e7b69fb36e4f47")]},  
{Name: "102", Date: "2021-04-12T18:25:43-05:00", Horizon: 262,  
Diameters_id: [ObjectId("61949fa714e7b69fb36e4f42")],  
Samples_id: [ObjectId("6194a0c314e7b69fb36e4f48")]}  
)
```

Для тестирования созданной БД выполним ряд запросов выборки данных.

А) Выберем пробы, номер которых содержит 24:

```
# db.Samples.find({Nomer: /24/i})
```

Б) Выведем таблицу проб вместе с документом «Описание», id которого указан в документе проб при помощи оператора \$lookup:

```
#db.Samples.aggregate([  
  $lookup: {  
    from: "Description",  
    localField: "Description_id",  
    foreignField: "_id",  
    as: "Description"  
  }  
])
```

В) Аналогично запросу Б, выведем таблицу скважин.

```
#db.Wells.aggregate([  
  $lookup: {  
    from: "Diameters",  
    localField: "Diameters_id",  
    foreignField: "_id",  
    as: "Diameters"  
  }  
],  
{ $lookup: {  
  from: "Samples",  
  localField: "Samples_id",
```

```
        foreignField: "_id",  
        as: "Samples"  
    }  
  ]])
```

Г) К предыдущему запросу В добавим фильтрацию. Выведем скважины с горизонтом меньше 260:

```
# db.Wells.aggregate([  
  { $match:  
    { Horizon: { $lt: 260 } }  
  },  
  { $lookup: {  
    from: "Diameters",  
    localField: "Diameters_id",  
    foreignField: "_id",  
    as: "Diameters"  
  }  
},  
  { $lookup: {  
    from: "Samples",  
    localField: "Samples_id",  
    foreignField: "_id",  
    as: "Samples"  
  }  
}])
```

Не смотря на то, что реляционные базы данных, основанные на мощном математическом аппарате, занимают доминирующее положение на рынке промышленных СУБД, развитие сферы хранения данных не стоит на месте. В этом заключается научный подход, связанный с постоянным обновлением знаний, представляющий собой циклический процесс [4]. В результате проделанной работы продемонстрирована возможность перехода к нереляционному подходу в области построения баз данных на примере нереляционной СУБД MongoDB и её использования в производственных процессах горной промышленности. Приведены примеры не только создания и заполнения базы данных «Скважины», но и выборки данных.

Список литературы

1. Шек, В.М. Открытые программные системы с применением геоинформационных технологий в горной промышленности / В.М. Шек, Е.А. Конкин // Программные продукты и системы. – 2007. – Т.1. – № 1. – С. 18–21.
2. Базы данных - GOSy VMKSS [Электронный ресурс]. – Режим доступа: <https://sites.google.com/site/gosyvmkss12/bazy-dannyh>. – Дата доступа: 14.03.2022.

3. Руководство по MongoDB. Преимущества – PROSELYTE [Электронный ресурс]. – Режим доступа: <https://proselyte.net/tutorials/mongodb/advantages/>. – Дата доступа: 13.02.2022.

4. Цимринг Д. Что такое наука, и как она работает / пер. с англ. В.С. Яценкова. Москва: ДМК Пресс, 2021. 326 с.

Куксевич Виталий Фёдорович, ст. преп. кафедры «Информационные системы и автоматизация производства», rallmall5@bk.ru, Республика Беларусь, Витебск, УО «Витебский государственный технологический университет».

Соколова Анна Сергеевна, ст. преп. кафедры «Информационные системы и автоматизация производства», Sokolova203509@gmail.com, Республика Беларусь, Витебск, УО «Витебский государственный технологический университет».

Черненко Дмитрий Владимирович, ст. преп. кафедры «Информационные системы и автоматизация производства», Chernenko203509@gmail.com, Республика Беларусь, Витебск, УО «Витебский государственный технологический университет».

USING MONGODB DBMS IN AUTOMATION OF MINING PROCESSES

Kuksevich V.F., Sokolova A.S., Chernenko D.V.

Annotation: the article presents the stages of the introduction of information technologies in the production processes of the mining industry and their development trends. A comparison of relational and document-oriented DBMS models was carried out, options for creating and filling a database in the MongoDB DBMS as applied to mining processes were considered.

Keywords: information technology, automation, mining, databases, non-relational DBMS.