Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра математического и компьютерного моделирования

Организация хранения и обработки информации об абитуриантах с помощью технологии NoSQL

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 451 группы

направления 38.03.05 - Бизнес-информатика

механико-математический факультет

Гусенковой Марии Сергеевны

Научный руководитель
доцент, к.ф. - м.н. О.М. Ромакина

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент Ю. А. Блинков

Саратов 2019

Введение. В современном мире стремительными темпами развиваются информационные технологии и те сферы человеческой деятельности, которые с ними связаны. Основные идеи современных информационных технологий базируются на концепции БД. Согласно данной концепции основой информационной технологии являются данные, организованные в БД, адекватно отражающие реалии действительности в той или иной предметной области и обеспечивающие пользователя актуальной информацией в соответствующей предметной области. В широком смысле слова база данных это совокупность описаний объектов предметной области и связей между ними, актуальных для конкретной предметной области.

Так же современная жизнь немыслима без эффективного управления. Важной категорией являются системы обработки информации, от которых во много зависит эффективность работы любого предприятия или учреждения.

Однако, работа с базами данных трудоемкая и утомительная. Для создания, введения и осуществления возможности коллективного пользования базами данных используются программные средства, называемые системами управления базами данных (СУБД). Система управления базами данных - это совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Существует несколько классификаций СУБД: по модели данных, по степени распределенности, по способу доступа к БД. Среди СУБД, классифицируемых по модели данных, выделяют: иерархические, сетевые, реляционные, объектно-ориентированные (документоориентированные), объектно-реляционные.

На практике наиболее используемы реляционные СУБД. Реляционные СУБД - СУБД, управляющие реляционными базами данных. Реляционная модель ориентирована на представление данных в виде двумерных таблиц. Такое представление удобно для пользователей. Но реляционные базы данных не могут справляться с нагрузками актуальными в наше время.

Это является главной причиной, которая заставляет разработчиков и бизнес приглядываться к альтернативам реляционных баз данных, используе-

мым вот уже более тридцати лет. В совокупности все эти технологии известны как «NoSQL базы данных».

Актуальность данной работы заключается в том, что в настоящее время количество неоднородных данных стремительно растет (по прогнозам IDG Research к 2022 г. 93% всех цифровых данных будут неоднородными), а это окажет существенное воздействие как на текущие, так и на будущие процессы управления данными. Поэтому необходимо создать такую информационную систему, которая позволит хранить большое количество структурированной и неструктурированной информации, а также эффективно использовать ее.

Данная бакалаврская работа посвящена документо-ориентированной базе данных MongoDB и многофункциональному языку Python.

Целью данной бакалаврской работы является организация хранения и обработки информации, предоставляемой абитуриентами при подачи документов в ВУЗ.

Основная часть. Основная часть работы содержит 4 раздела:

1. Тенологии разработки баз данных;
2. Документо-ориентированная база данных MongoDB;
3. Многофункциональный язык программирования Python;
4. Разработка концепции информационной системы «Абитуриент».

В первом разделе приводится краткое описания основных понятий баз данных и описание технологии NoSQL.

База данных – это совокупность связанных данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, независимая от прикладных программ. База данных является информационной моделью предметной области. Обращение к базам данных осуществляется с помощью системы управления базами данных (СУБД). СУБД обеспечивает поддержку создания баз данных, централизованного управления и организации доступа к ним различных пользователей.

Основными конструктивными элементами модели хранения данных в БД является:

- сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных.
- атрибут – поименованная характеристика сущности.
- ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.
- связь – ассоциирование двух или более сущностей.

СУБД бывают двух видов: реалиционные и нереалиционные.

Нереалиционные СУБД или технология NoSQL - это термин, обозначающий ряд подходов, направленных на реализацию хранилищ баз данных, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL.

Технологии NoSQL появились при попытке решить проблему реалиционных баз данных, а именно горизонтальную масштабируемость, то есть невозможность наращивать производительность путём добавления новых вычислительных узлов к уже работающим.

Поэтому большинству NoSQL-систем имеют распределённую архитектуру, которая решает проблему горизонтальной масштабируемости и увеличивает надежность системы с помощью поддержания нескольких копий данных.

Технологии NoSQL не подразумевают внутренних связей, которые позволяют снять ограничения с формирования сущностей и допускают хранения данных в виде ключ-значение.

Если реалиционные СУБД базируются на требованиях ACID к транзакционной системе: атомарности, согласованности, изолированности, надежности, то NoSQL ориентируется на наборе свойств BASE:

- базовая доступность, т.е каждый запрос гарантированно завершается (успешно или безуспешно);
- гибкое состояние, т.е состояние системы может изменяться со временем, даже без ввода новых данных, для достижения согласования данных;
- согласованность в конечном счете, т,е данные могут быть некоторое время рассогласованы, но приходят к согласованию через некоторое время.

Технологии NoSQL присуще:

- Применение различных типов хранилищ.

- Возможность разработки базы данных без задания схемы.
- Использование многопроцессорности.
- Линейная масштабируемость (добавление процессоров увеличивает производительность).
- Инновационность: «не только SQL» открывает много возможностей для хранения и обработки данных.
- Сокращение времени разработки.
- Скорость: даже при небольшом количестве данных конечные пользователи могут оценить снижение времени отклика системы с сотен миллисекунд до миллисекунд.

Технология NoSQL обладает нереляционными моделями данными, которые значительно проще, чем модели классической реляционной модели, и свои способы для осуществления запросов, которые существенно отличаются от традиционных баз данных.

Основываясь на модели данных и подходах к распределенности и репликации можно выделить основные классы технологии NoSQL:

1. Системы «ключ-значение» (Key-Value Stores).
2. Документно-ориентированые (Document Stores).
3. Системы хранилищ семейств колонок (Extensible Record Stores / Wide Column Stores / Column Families).
4. Графовые базы данных.

Данное деление достаточно общее, так как внутри каждой группы системы значительно различаются и в плане поддержки согласованности данных, и в нюансах работы с ними.

Обычно многие системы обладают свойствами более чем одного класса, и иногда их трудно классифицировать по такому принципу. Далее приведена краткая характеристика каждого класса систем.

Системы «ключ-значение»

В данном типе NoSQL-систем хранятся как структурированные, так и неструктурированные данные и доступ к ним осуществляется при помощи уникального ключа, который является единственным, без поддержки вторичных ключей и индексов. Так же здесь может поддерживаться некоторая

структура данных, позволяющая менять отдельные поля объекта, но не позволяющая строить по ним запросы.

Работа с данными обычно осуществляется с помощью простых операций вставки, удаления и поиска по ключу. В этом отношении системы «ключ-значение» похожи на популярную распределённую систему кэширования в оперативной памяти Memcached, но предоставляют постоянное хранение данных и ряд дополнительных возможностей.

Документо-ориентированные базы данных

В отличие от систем класса «ключ-значение» у документо-ориентированные СУБД предоставляют намного больше возможностей. Еще одним отличием документо-ориентированные СУБД от систем типа «ключ-значение» является то, что документные СУБД могут запрашивать коллекции документов на основании нескольких ограничений на атрибуты, могут осуществлять агрегатные запросы, сортировку результатов, поддерживают индексы на полях документов и т.д. Также документно-ориентированные системы поддерживают поиск по полям документов, индексы, часто допускаются вложенные документы и массивы, при этом схемы данных,которая предопределена заранее, нет.

Единицей хранения данных в таких системах является документ – некоторый объект, обладающий произвольным набором атрибутов (полей), который может быть представлен, например, в JSON.

Документы могут быть организованы (сгруппированы) в коллекции. Их можно считать отдаленным аналогом таблиц реляционных СУБД, но коллекции могут содержать другие коллекции. Хотя документы коллекции могут быть произвольными, для более эффективного индексирования лучше объединять в коллекцию документы с похожей структурой. Документо-ориентированные базы данных применяются в системах управления содержимым, издательском деле, документальном поиске и т. п.

Системы хранилищ семейств колонок

Основной идей такого класса систем NoSQL является хранение данных не по строкам, как это принято в реляционных СУБД, а по колонкам. Это означает, что с точки зрения пользователя данные представлены как обычно в виде таблиц, но физически эти таблицы являются совокупностью колонок,

каждая из которых по сути представляет собой таблицу из одного поля. Физически эти данные хранятся последовательно друг другу.

Такая структура хранения данных означает, что при выполнении запроса на чтение, в котором фигурируют только 2 поля из 20 полей таблицы, реально будут прочитаны только 2 колонки.

Это означает что нагрузка на канал ввода/вывода будет приблизительно в 10 раз меньше чем при выполнении такого же запроса в реляционной СУБД.

При хранении данных в таком классе систем NoSQL появляется возможность компрессии данные, так как в одной колонке таблицы данные, как правило, однотипные.

Графовые базы данных

Графовая база данных – это база данных, предназначенная для хранения данных в виде графа. Т.е. это хранения данных по вершинам и ребрам.

По определению, графовая база данных – это любая структура для хранения, где взаимосвязанные элементы соединены без применения индекса. Смежные по структуре элементы доступны при разыменовании физического показателя. Существует несколько типов графов, которые могут хранить: от «однотипного» ненаправленного графа до гиперграфа, включая свойственные подграфы.

Таким образом, база данных в виде графа должна соответствовать следующим критериям:

- хранение оптимизировано для данных, представленных в виде графа, с обеспечением хранения данных по вершинам и ребрам;
- хранение оптимизировано для просмотра графа без использования индекса.

  Графовая база данных оптимизирована для запросов используя близость данных, начиная от одного или нескольких корневых узлов, а не использование глобальных запросов;
- гибкая модель данных для некоторых решений: нет необходимости в определении типов данных для вершин и ребер в отличие от более ограниченной табличной модели реляционной базы данных;
- встроенная API с точкой входа для классических алгоритмов теории графов (кратчайший путь, алгоритм Дейкстры, A* и др.)

Во втором разделе описывается документо-ориентированная база данных MongoDB.

MongoDB принадлежит к типу документо-ориентированных баз данных, которые характеризуются тем, что каждая запись является документом, у которого отсутсвует жестко заданная схема, который также может содержать вложенные документы. MongoDB реализует новый подход к построению баз данных, где нет таблиц, схем, запросов SQL, внешних ключей и многих других вещей, которые присущи объектно-реляционным базам данных.

Вся система MongoDB может представлять не только одну базу данных, находящуюся на одном физическом сервере. Функциональность MongoDB позволяет расположить несколько баз данных на нескольких физических серверах, и эти базы данных смогут легко обмениваться данными и сохранять целостность. Одним из популярных стандартов обмена данными и их хранения является JSON. JSON эффективно описывает сложные по структуре данные. Для хранения в MongoDB применяется формат, который называется BSON или сокращение от binary JSON. BSON позволяет работать с данными быстрее: быстрее выполняется поиск и обработка. Хотя надо отметить, что BSON в отличие от хранения данных в формате JSON имеет небольшой недостаток: в целом данные в JSON-формате занимают меньше места, чем в формате BSON, но данный недостаток компенсируется высокой скоростью обработки данных. Система хранения данных в MongoDB представляет набор реплик. В этом наборе есть основной узел, а также может быть набор вторичных узлов. Все вторичные узлы сохраняют целостность и автоматически обновляются вместе с обновлением главного узла. И если основной узел по каким-то причинам выходит из строя, то один из вторичных узлов становится главным. Отсутствие жесткой схемы базы данных и в связи с этим потребности при малейшем изменении концепции хранения данных пересоздавать эту схему значительно облегчают работу с базами данных MongoDB и дальнейшим их масштабированием, что значительно экономит время на построение сложных запросов. В результате мы можем выделить ряд основных возможностей СУБД MongoDB:

- Документо-ориентированное хранилище (простая и мощная JSON- подобная схема данных);

- Достаточно гибкий язык для формирования запросов;
- Динамические запросы;
- Полная поддержка индексов;
- Профилирование запросов;
- Быстрые обновления «на месте»;
- Эффективное хранение двоичных данных больших объемов (фото и видео);
- Журналирование операций, модифицирующих данные в БД;
- Поддержка отказоустойчивости и масштабируемости: асинхронная репликация, набор реплик и шардинг;
- Может работать в соответствии с парадигмой MapReduce;

Также следует отметить, что наряду с достоинствами система управления базами данных имеет ряд недостатков:

- Отсутствие оператора «join». Обычно данные могут быть организованы более денормализованным способом, но на разработчиков ложится дополнительная нагрузка по обеспечению непротиворечивости данных.
- Отсутствие понятия «транзакции». Атомарность гарантируется только на уровне целого документа, т.е. частичное обновление документа произойти не может.
- Отсутствует понятия «изоляции». Любые данные, которые считываются одним клиентом, могут параллельно изменять- ся другим клиентом. Одним из важнейших аспектов является то, что клиентские драйверы MongoDB поддерживают все популярные языки программирования, а это значит, что разработчику предоставляется возможность конструирования запросов на любом удобном для него языке.

В третьем разделе представлена информация о многофункциональном языке программирования Python.

Python высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное

и аспектноориентированное. Основные архитектурные черты динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных.

Преимущества языка являются:

1. Интерпретируемость.
2. Динамическая типизация.
3. Хорошая поддержка модульности.
4. Встроенная поддержка Unicode в строках.
5. Поддержка объектно-ориентированного программирования.
6. Автоматическая сборка мусора, отсутствие утечек памяти.
7. Интеграция с C/C++.
8. Понятный и лаконичный синтаксис, способствующий ясному отображению кода.
9. Огромное количество модулей, как входящих в стандартную поставку Python 3, так и сторонних.
10. Кроссплатформенность.

Язык программирования Python прост, универсален и практически ничем не ограничен.

Четвертый раздел посвящен разработке концепции информационной системы «Абитуриент».

На основе Федерального закона от 22.10.2004 N 125-ФЗ (ред. от 28.12.2017) «Об архивном деле в Российской Федерации» ВУЗ должен хранить данные, предоставляемые абитуриентами в ходе приемной комиссии, в течении 75 лет. Хранить данные в бумажном варианте не очень практично и удобно, поэтому возникает необходимость создать информационную систему. Так как данные неструктурированные , то необходимо создать информационную систему при помощи технологии NoSQL.

Изначально данные, предоставляемые абитуриентами, были представлены в виде плохо структурированных excel-файлов. Для обработки данной информации был создан скрипт, который автоматически обрабатывает excel-файлы, находящиеся в одном каталоге вместе с данным скриптом и создает документы формата JSON.

Также был написан скрипт по созданию JSON-схем, необходимость которого обусловлено тем, что JSON-схемы используются для валидации документов JSON формата, каждый из которых имееет свою собственную, отличную от других, структуру.

Также в данном разделе была написана программа для создания базы данных под управлением MongoDB, которая позволяет удобно хранить и обрабатывать структурированную и неструктурированную информацию, которая предоставляется абитуриентами в ходе подачи документов при поступлении в ВУЗ. Также к ссозданной базе данных были написаны типовые запросы для демонстрации работы базы данных.

Заключение. MongoDB представляет собой удобную для работы с большими объемами данных open source систему баз данных, которая не требует какого-либо описания схемы базы данных и может быть быстро расширена (при помощи кода в формате JSON) по сравнению с конкурирующими системами она может постепенно меняться по мере развития приложения, что представляет большое удобство для разработчика.

Так же СУБД MongoDB является достаточно перспективной системой управления базами данных. Однако, не смотря на множество ее преимуществ, система имеет существенные недостатки, так как данная СУБД не подходит для исполнения сложных и многокомпонентных задач.

При выполнении данной работы в теоретической части были рассмотрены важнейшие подходы, методы, структуры разработки и проектирования нереляционных баз данных, проведено аналитическое сравнение различных платформ работы с NoSQL. Так же были рассмотрены теоритические сведения о документо-ориентированной модели данных MongoDB, о формате представления данных в MongoDB (JSON) и формате хранения данных в MongoDB (BSON) и о высокоуровневом, многофункциональном языке программирования Python.

В практической части данной работы были реализованы программы по:

- обработке excel-файлов, содержащих информацию об абитуриентах;
- созданию на основе извлеченной информации JSON-документов;
- созданию на основе уже имеющихся JSON файлов JSON-схемы, полностью описывающие их.

- созданию документо-ориентированной базы данных для хранения и обработки информации предоставляемой абитуриентами при подаче документов для поступления в ВУЗ под управлением MongoDB.
- реализации комплекса типовых запросов к документо-ориентированной базе данных MongoDB.