**Отчёт.**

**База данных LiteDB.**

***LiteDB*** - кроссплатформенная нереляционная база данных с открытым исходным кодом. LiteDB представляет собой бессерверную базу данных, поставляемую в виде одной небольшой DLL. LiteDB полностью написана на C#, что дает ей возможность работать на любой платформе, поддерживающей .NET Framework. LiteDB - простая, быстрая и легкая встроенная база данных. Создатели LiteDB были вдохновлены документарной бд MongoDB (которую мы проходили пару месяцев назад), поэтому ее API очень похож на официальный .NET API MongoDB. Разработчики рекомендуют использовать LiteDB для мобильных, локальных приложений и веб-сервисов.

⦁ ***История развития СУБД:***

В 2012-2013 годах у главного разработчика, профессора университета Сан-Маркуса, возникла идея о создании упрощенной, удобной и open-source версии набиравшей популярность и небезызвестной нам MongoDB. В 2013 год он и несколько коллег по работе решили начать совместную разработку грандиозного для малой группы проекта. Тем не менее благодаря участию и помощи добровольцев среди студентов и других, уже через год LiteDB вышла в свет: 14 августа 2014 года появился оф. сайт, на котором объявили о запуске проекта СУБД и выложили первую версию. Также для пользователей были сразу обозначены и принципы, и конечная цель проекта. Постепенно, вместе с увеличением команды росло и число возможностей новой СУБД: в среднем фиксы и дополнения выходили ежемесячно, а глобальные обновления - ежегодно. Поначалу, работа с LiteDB была возможна только с помощью .NET Framework и Visual Studio (используя C#), однако уже к 2016 году появилась и поддержка языка, крайне близкого к SQL. Вместе с этим нововведением команда обнародавала собственную LiteDB.Studio, предназначенную для упрощенной работы на системах Windows на языке SQL. Через год были добавлены транзакции и обновлена система обработки запросов на SQL. В 2018 году поставлен новый движок (WAL), кроме того появилась возможность контроля мультипроцессинга и многопоточности. Далее из крупных обновлений, к сожалению, имело место лишь глобальное изменение Debugger-а, добавление функции Ensure. Сейчас СУБД все еще развивается, а в ближайшее время ожидается крупное обновление LiteDB.Studio.

***⦁*** ***Инструменты для взаимодействия с СУБД:***

Может работать на любой платформе, поддерживающей .Net Framework, является кроссплатформенной. Имеется Studio - очень удобная для Windows.

***⦁*** ***Какой database engine используется в вашей СУБД?***

В LiteDB используются два “движка”: PragmaS Engine и WAL (Write Ahead Log) LiteDB Engine.

***⦁*** ***Распределение файлов БД по разным носителям?***

LiteDB является бессерверной, на одного пользователя/аккаунт рассчитана всего одна база данных и один файл данных (как в SQLite).

***⦁*** ***На каком языке/ах программирования написана СУБД?***

Как уже говорилось ранее, LiteDB полностью написана на C#

***⦁*** ***Какие типы индексов поддерживаются в БД? Приведите пример создания индексов.***

Поддерживаются уникальные, составные и мультиключевые индексы.

***// Создание мультиключевого индекса в телефонных номерах***

***col.EnsureIndex(x => x.Phones, "$.Phones[\*]");***

***⦁*** ***Как строится процесс выполнения запросов в вашей СУБД?***

Пустая страница (Empty Page): когда страница любого типа удаляется из списка, она становится пустой страницей. Эта пустая страница будет использована при запросе новой страницы и преобразуется к требуемому типу. Далее ищется первая подходящая страница (одна из Header Page, Collection Page, Index Page, Data Page, Extend Page, Empty Page). Например, для запросов по типу записи есть такая схема: в БД постоянно поддерживается последовательность страниц по убыванию свободного места, в первую записывается Ваша информация, и как только место кончится, страница удалится из списка и Вы перейдете к следующей. С другими запросами всё даже проще.

***⦁*** ***Как устроен язык запросов в вашей СУБД? Разверните БД с данными и выполните ряд запросов.***

***+*** ***Есть ли для вашей СУБД понятие «план запросов»? Если да, объясните, как работает данный этап.***

В Студии - все аналогично Postgres, а в стандартной версии доступна возможность записать несколько запросов в одной группе, где они будут выполняться последовательно/по приоритетам:

***using(var db = new LiteDatabase(@"C:\Temp\MyData.db"))***

***{***

***// Получаем коллекцию***

***var col = db.GetCollection<Customer>("customers");***

***// Создаем новый экземпляр класса***

***var customer = new Customer***

***{***

***Name = "John Doe",***

***Phones = new string[] { "8000-0000", "9000-0000" },***

***IsActive = true***

***};***

***// Вставить новый документ (идентификатор будет автоматически увеличен)***

***col.Insert(customer);***

***// Обновить документ внутри коллекции***

***customer.Name = "Joana Doe";***

***col.Update(customer);***

***// Индексировать документ, используя свойство документа Name***

***col.EnsureIndex(x => x.Name);***

***// Использовать LINQ для запроса документов***

***var results = col.Find(x => x.Name.StartsWith("Jo"));***

***// Создание мультиключевого индекса в телефонных номерах***

***col.EnsureIndex(x => x.Phones, "$.Phones[\*]");***

***// и теперь мы можем запросить телефоны***

***var r = col.FindOne(x => x.Phones.Contains("8888-5555"));***

***}***

***⦁*** ***Поддерживаются ли транзакции в вашей СУБД? Если да, то расскажите о нем. Если нет, то существует ли альтернатива?***

Да, поддерживаются, соответствуют принципам ACID. Несмотря на тяжелую историю развития транзакций в БД, они все же были добавлены в 2018 году, причем со специальным “Transaction monitor” для управления и распределения памяти на страницах при использовании больших multi-transactions. Реализация транзакций в БД идейно не отличается от MongoDB или Postgres в Студии, основными командами для работы с ними являются: begin, commit and rollback

***⦁*** ***Какие методы восстановления поддерживаются в вашей СУБД. Расскажите о них.***

Восстановление данных при сбое записи (WAL - Write Ahead Log mode); возможность восстановления по образу коллекции в Secured Header Page.

***⦁*** ***Расскажите про шардинг в вашей конкретной СУБД. Какие типы используются? Принцип работы.***

Можно считать, что он вообще отсутствует (как я понял, допустимо лишь использование нескольких ПК локально). LiteDB рассчитана для небольших мобильных и локальных приложений, а также для веб-сервисов без больших аудиторий. В принципе - логично, она же ***Lite.***

***⦁*** ***Возможно ли применить термины Data Mining, Data Warehousing и OLAP в вашей СУБД?***

К сожалению, БД - любительская и ограничена в функционале. Однако, учитывая прогресс небольшой команды разработчиков, не удивлюсь, если увижу подобное в будущих версиях.

***⦁*** ***Какие методы защиты поддерживаются вашей СУБД? Шифрование трафика, модели авторизации и т.п.***

Имеется возможность шифрования файлов и коллекции (DES cryptography). Более того, предусмотрена возможность “блокировки свободного доступа” к отдельным страницам, например, Header Page, с помощью команды на C# при обращении к странице/работе с памятью или сразу при создании/обновлении коллекции. Вследствие бессерверности необходимости в защите “онлайн” - попросту нет.

***⦁*** ***Какие сообщества развивают данную СУБД? Кто в проекте имеет права на коммит и создание дистрибутива версий? Расскажите об этих людей и/или компаниях.***

Идея создания рассматриваемой СУБД принадлежит Давиду Маурицио, гражданину Перу и по совместительству профессору престижнейшего и старейшего университета страны:  
<https://www.researchgate.net/profile/David-Mauricio>

Давидом в 2013 году была основана команда LiteDB Team (кто бы мог подумать), которая фокусировалась на создании одноименной БД и выпустила ее в Августе 2014-го. Совершенствование СУБД продолжается и самой командой, и при помощи “добровольцев”: так, последняя версия была загружена на оф. сайт и гитхаб чуть менее двух лет назад. Она стала шестой с выпуска.

Это open-source проект: пользование БД не требует платы и даже больших стараний при установке, не требует регистрации. Тем не менее, гитхаб разработчика и оф. сайт редактируются только компанией и создателем лично. Стоит отметить, что все пожилые версии сохранены на гитхабе и при желании пользователей могут быть использованы свободно.

Справка об авторе и команде:

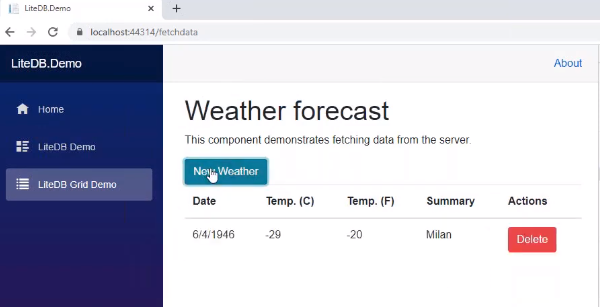
Команда в почти том же составе не раз участвовала в совместных разнообразных проектах, работая как и на государство (в военной отрасли и в отрасли создания пром. ПО), так и в собственных проектах по типу той же LiteDB. Интересно, что команда кроме того сотрудничает с известнейшим университетом MIT. Давид Маурицио является автором более 80 научных публикаций, другие лидеры его команды также не отстают.

***⦁*** ***Создайте свои собственные данные для демонстрации работы СУБД.***

***⦁*** ***Как продолжить самостоятельное изучение языка запросов с помощью демобазы. Если демобазы нет, то создайте ее.***

Демобаза - есть! Она называется LiteDB.Demo, установить и почитать о ней можно по ссылке:

<https://github.com/andreatosato/LiteDBSamples>

Интерфейс я бы назвал крайне удобным и понятным, низкая нагруженность как бы “подсказывает” пользователю:******

***⦁*** ***Где найти документацию и пройти обучение***

Несмотря на сравнительно небольшую популярность, у БД было легко найти и оф. документацию, и гайды:

<https://www.litedb.org/> - оф. сайт

<https://github.com/mbdavid/LiteDB/wiki/How-LiteDB-Works/> - документация

<https://github.com/mbdavid/LiteDB/wiki/Getting-Started/> - гайд для новичка

***⦁*** ***Как быть в курсе происходящего***

Следить за оф. сайтом и удобным гитхабом главного разработчика.