

1. История развития СУБД:

BaseX был создан в 2005 году Кристианом Грюном, научным сотрудником [Констанцского университета](#) (Германия). С 2007 года BaseX распространяется в открытых исходных текстах.

2. Инструменты для взаимодействия с СУБД

APIs: RESTXQ, [REST](#), [WebDAV](#), XML:DB, XQJ;

Языки программирования:

- Actionscript
- C
- C#
- Haskell
- Java
- JavaScript
- Lisp
- Perl
- PHP
- Python
- Qt
- Rebol
- Ruby
- Scala
- Visual Basic

3. Какой database engine используется в вашей СУБД?

XQuery

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

Есть 3 варианта запросов к БД: Find, XQuery, Command. Основной XQuery, остальное – надстройки, которые переводят запрос на язык XQuery

Find

In the **Find** mode, the input bar can be used to find single elements and texts in the currently opened database. The following syntax is supported:

Query	Description
city	Find elements named <code>city</code> , and texts containing this token.
=India	Find texts matching the exact string <code>India</code> .
~Cingdom	Find texts equal or similar to the token <code>Cingdom</code> .
@id	Find attributes named <code>id</code> and attribute values containing this token.
@=f0_119	Find attribute values matching the exact string <code>f0_119</code> .
"European Chinese"	Find texts containing the phrase <code>"European Chinese"</code> .
//city	Leading slash: Interpret the input as XPath expression (see below).

Запрашиваем информацию по России:

Query: **countryName=Russia**

C:\Program Files (x86)\J...
Find contents...

1

OK

factbook.xml

Content: db... Editor

1 Result, 40 KB

```

<country id="f0_358" name="Russia" capital="f0_1598" population="148178480" datacode="RS" total_area="17075200" population_growth="-0.87" infant_mortality="24.7" gdp_apri="6" gdp_total="796000" inflation="7" indep_date="24 08 1991" government="federation" gdp_inde="41" gdp_serv="53" car_code="R">
  <nameRussia/name>
    <city id="f0_5805" country="f0_358">
      <nameSergiev Posad/name>
        <nameZagorsk/name>
          <population year="95">114000</population>
        </city>
        <city id="f0_5847" country="f0_358">
          <nameSeversk/name>
            <population year="95">110000</population>
          </city>
        <province id="f0_17933" name="Rep. of Karelia" country="f0_358" capital="f0_4239" population="785000" area="172480">
          <city id="f0_4239" country="f0_358" province="f0_17933">
            <namePetrozavodsk/name>
              <population year="95">280000</population>
            </city>
            <province id="f0_17935" name="Rep. of Komi" country="f0_358" capital="f0_4245" population="1185500" area="415900">
              <city id="f0_4245" country="f0_358" province="f0_17935">
                <nameSyktyvkar/name>
                  <population year="95">122000</population>
                </city>
                <city id="f0_5102" country="f0_358" province="f0_17935">
                  <nameUdhta/name>
                    <population year="95">106000</population>
                  </city>
                  <city id="f0_5112" country="f0_358" province="f0_17935">
                    <nameVorkuta/name>
                      <population year="95">104000</population>
                    </city>

```

Result

```

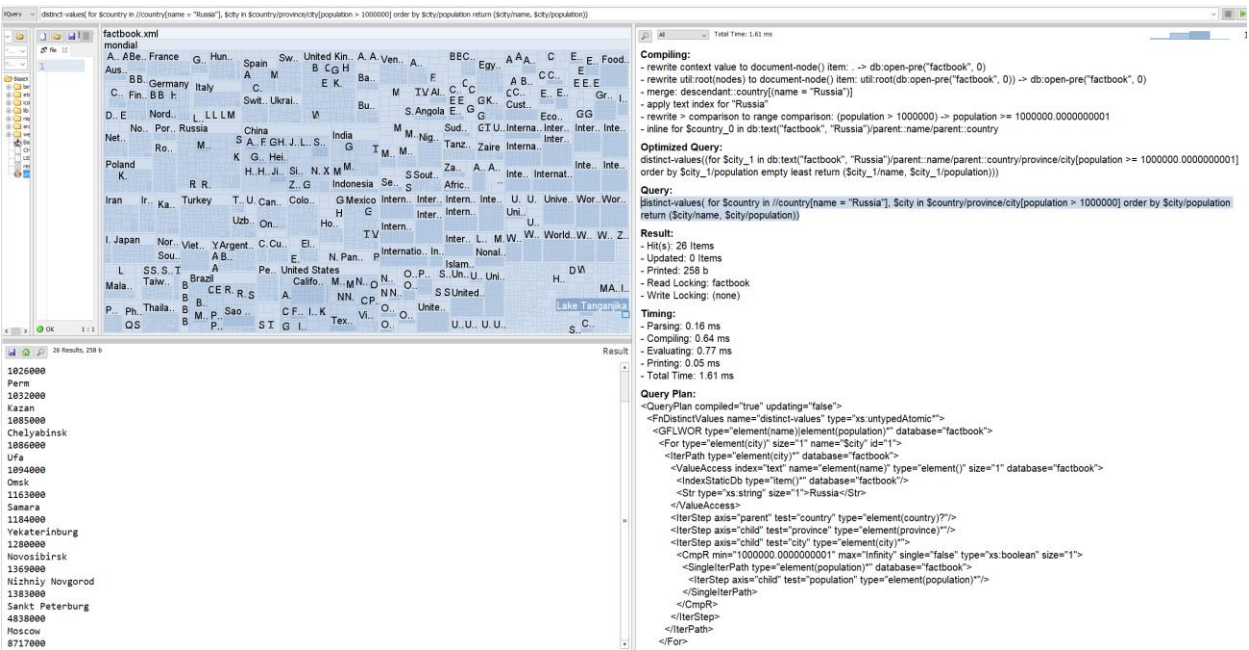
<QueryPlan>
  <QueryPlan compiled="true" updating="false">
    <IterPath type="document(country?) database="factbook">
      <ValueAccess index="text" name="element(name)" type="element"? size="1" database="factbook">
        <IndexStaticDb type="item"? database="factbook">
          <Str type="xs:string" size="1" value="Russia"/>
        </ValueAccess>
        <IterStep axis="parent" test="country?" type="element(country)?">
          <IterPath>

```

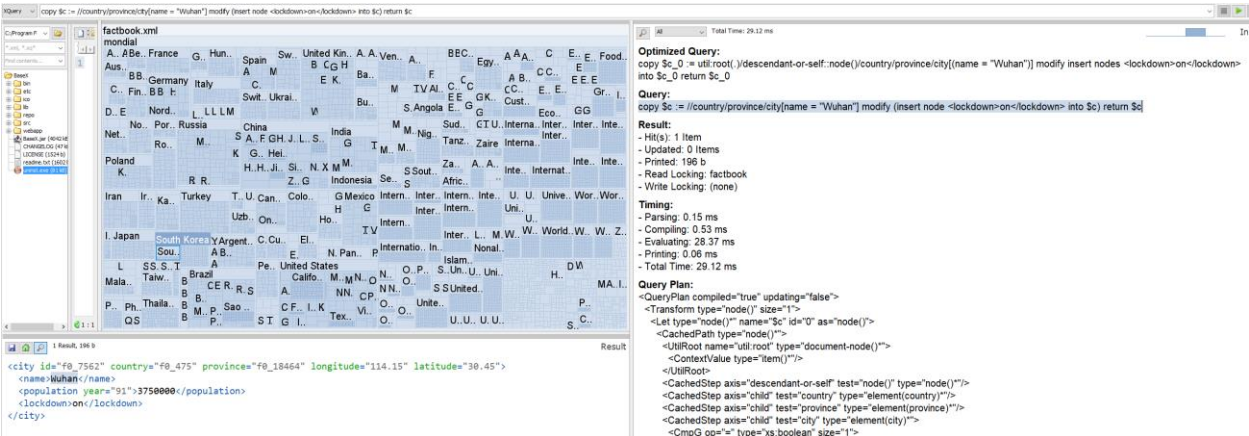
Запрашиваем информацию по городам миллионникам в лексикографическом порядке:

[illegible]

Запрашиваем информацию по российским городам миллионникам в порядке возрастания численности населения:



Добавляем городу Ухань информацию о локдауне:



5. Распределение файлов БД по разным носителям?

Нет

6. На каком языке/ах программирования написана СУБД?

Java

7. Какие типы индексов поддерживаются в БД? Приведите пример создания индексов.

Name Index (system)	Name Index содержит ссылки на имена всех элементов и атрибутов в базе данных. Он содержит некоторую базовую статистическую информацию, такую как частоту использования имени
Path Index (system)	Path Index хранит все уникальные пути к документам в базе данных. Он содержит дополнительную статистическую информацию, такую как частоту использования пути, его различные строковые значения и

	минимальное/максимальное количество числовых значений
Document Index (system)	Document Index содержит ссылки на все узлы документа в базе данных
Text Index	Этот индекс ссылается на текстовые узлы документов. Он будет использоваться для ускорения сравнения строк в пути
Attribute Index	Подобно Text Index, этот индекс ускоряет сравнение строк и диапазонов значений атрибутов
Token Index	Во многих диалектах XML, таких как HTML или DITA, множество токенов хранится в значениях атрибутов. Индекс токена может быть создан для ускорения поиска этих токенов
Full-Text Index	Full-Text Index содержит нормализованные маркеры текстовых узлов документа. Он используется для ускорения запросов с помощью текстового выражения <code>contains_text()</code> , и он способен обрабатывать операции <code>wildcard</code> и нечеткого поиска
Custom Index	С помощью XQuery сравнительно легко создавать свои собственные структуры индексов

Пример создания индекса по правительству в стране:

Command:

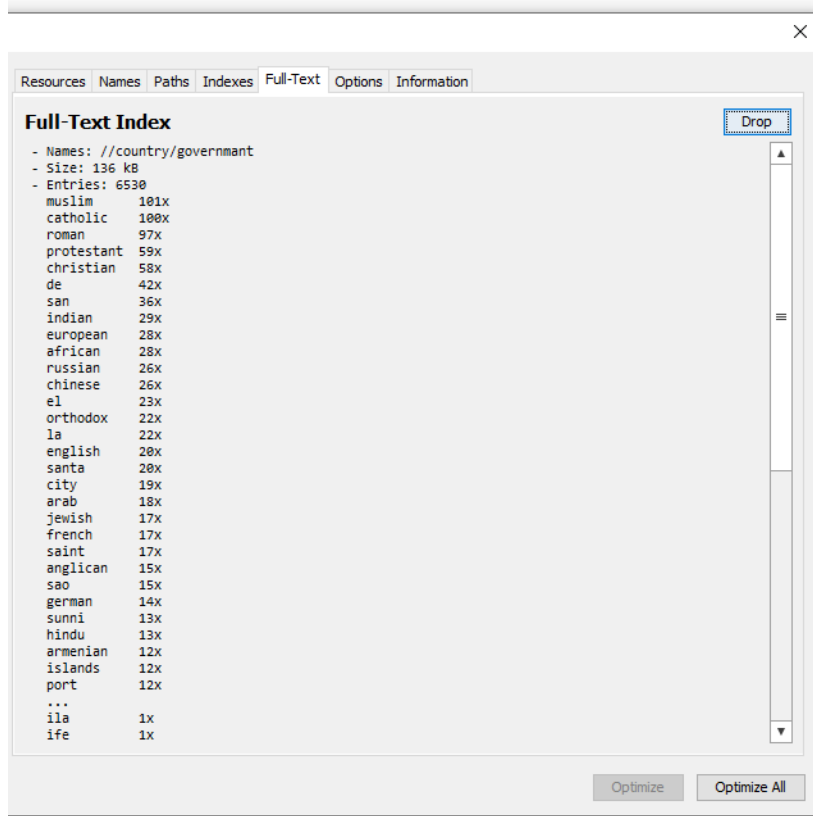
SET FTINCLUDE //country/government

Command:

CREATE INDEX fulltext

Result:

Index 'fulltext' created in 21.31 ms.



8. Как строится процесс выполнения запросов в вашей СУБД?

Запрос XQuery вычисляется в несколько этапов:

1. Во время синтаксического анализа строка запроса – основной модуль XQuery – преобразуется в древовидное представление, абстрактное синтаксическое дерево (AST).
2. Во время компиляции синтаксическое дерево украшается дополнительной информацией (информация о типе, свойства выражения); выражения перемещаются, упрощаются или предварительно вычисляются:
 1. Логические оптимизации не зависят от контекста. Они могут быть применены независимо от того, какие данные будут обрабатываться позже.
 2. Физическая оптимизация основана на контекстной информации, такой как статистика базы данных или доступные индексы.
3. Во время вычисления обрабатывается полученное дерево выражений.
4. Результаты возвращаются пользователю. Некоторые выражения (например, простые циклы) могут быть вычислены итеративным способом, в то время как другие (например, операции сортировки) должны быть полностью вычислены до того, как будет доступен первый результат.

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

Да, есть Query Plan:

Query Plan:

```
<QueryPlan compiled="true" updating="false">
  <Transform type="node()" size="1">
    <Let type="node()" name="$c" id="0" as="node()">
      <CachedPath type="node()">
        <UtilRoot name="util:root" type="document-node()">
          <ContextValue type="item()">
            </UtilRoot>
            <CachedStep axis="descendant-or-self" test="node()" type="node()">
              <CachedStep axis="child" test="country" type="element(country)">
                <CachedStep axis="child" test="province" type="element(province)">
                  <CachedStep axis="child" test="city" type="element(city)">
                    <CmpG op="=" type="xs:boolean" size="1">
                      <CachedPath type="node()">
                        <CachedStep axis="child" test="name" type="element(name)">
                          </CachedPath>
                          <Str type="xs:string" size="1">Wuhan</Str>
                        </CachedStep>
                      </CachedPath>
                    </CmpG>
                  </CachedStep>
                </CachedPath>
              </Let>
            <Insert type="empty-sequence()" size="0">
              <VarRef type="node()" size="1" name="$c" id="0" as="node()">
                <CElem type="element(lockdown)" size="1">
                  <QNm type="xs:QName" size="1">lockdown</QNm>
                  <Str type="xs:string" size="1">on</Str>
                </CElem>
              </Insert>
            <VarRef type="node()" size="1" name="$c" id="0" as="node()">
          </Transform>
        </QueryPlan>
```

Он создается во время компиляции XQuery запроса, с учетом логической и физической оптимизаций (в том числе на основе созданных индексов). Включает в себя дополнительную информацию об используемых типах и конфигурации запроса и представляет из себя готовую для выполнения движком инструкцию.

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

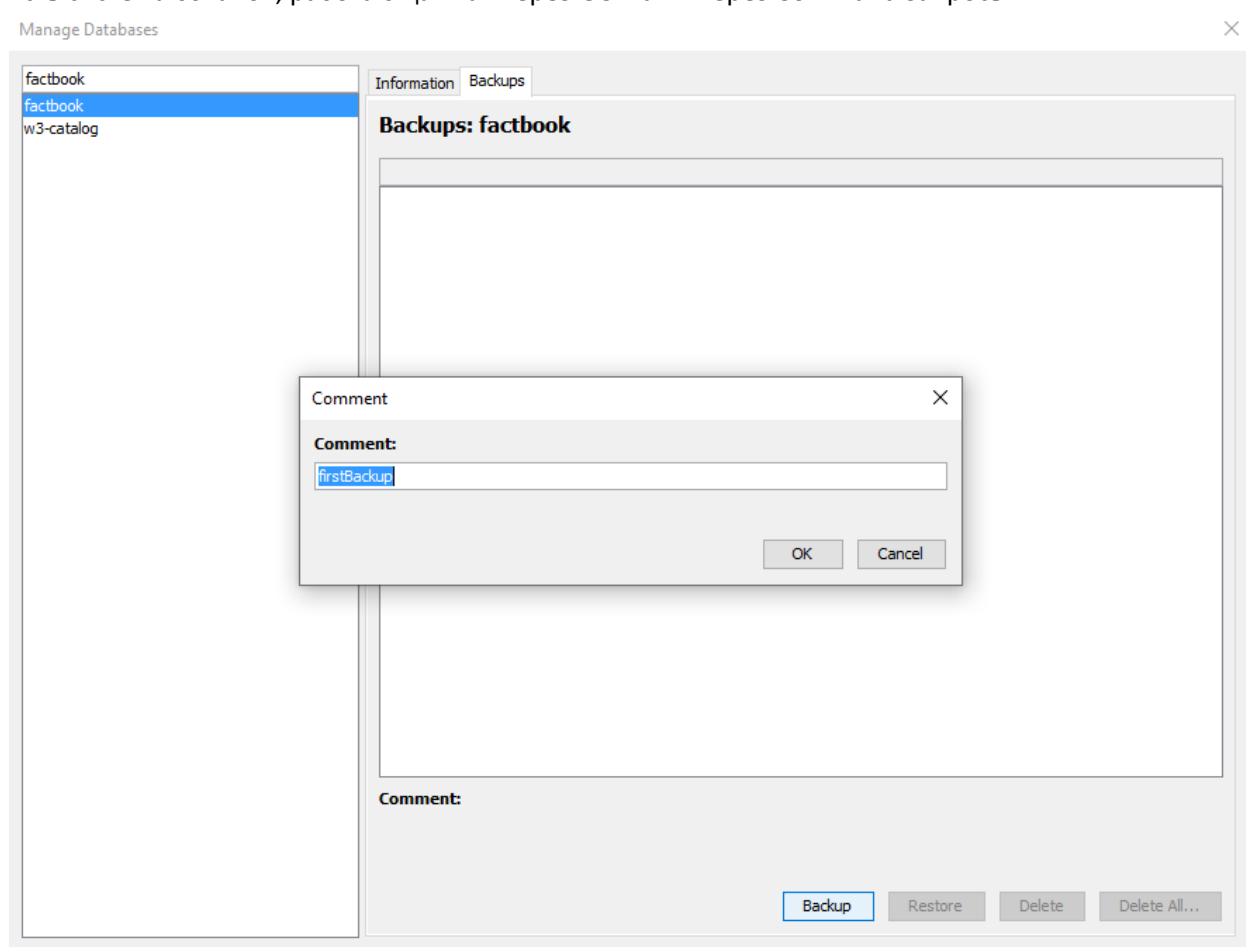
В двух словах, транзакция равна команде или запросу. Таким образом, каждая команда или запрос, отправленные на сервер, становятся транзакцией.

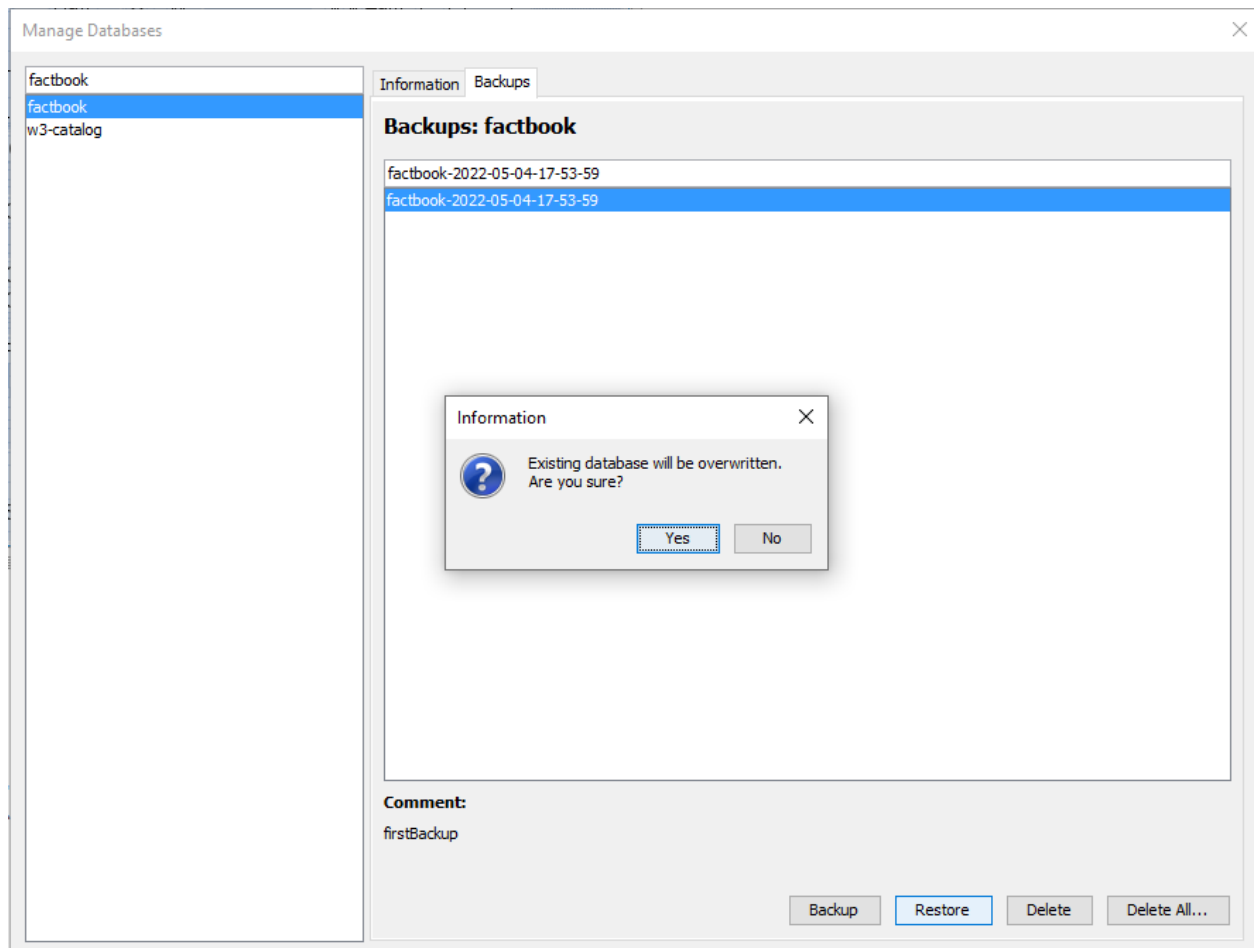
BaseX обеспечивает поддержку множественных операций чтения и одиночной записи (с использованием *proclaiming* и *starvation-free* двухфазной блокировки). Это означает, что:

- Транзакции чтения выполняются параллельно.
- Если поступает транзакция обновления, она будет помещена в очередь и выполнена после выполнения всех предыдущих транзакций чтения.
- Последующие операции (чтение или запись) будут помещены в очередь до завершения транзакции обновления.
- Задания без доступа к базе данных никогда не будут заблокированы. Задания с глобальной блокировкой теперь могут выполняться параллельно с заданиями без блокировки.
- Каждая база данных имеет свою собственную очередь: обновление базы данных А не блокирует операции с базой данных В. Это делается при условии, что это может быть определено статически, т.е. до оценки транзакции, к каким базам данных будет осуществляться доступ с помощью транзакции
- Максимальное количество параллельных транзакций можно настроить с помощью опции *PARALLEL*.
- По умолчанию предпочтение отдается транзакциям чтения, и транзакции, не имеющие доступа к базам данных, могут быть обработаны, даже если достигнут лимит транзакций. Это поведение можно изменить с помощью опции *AIRLOCK*.

11. Какие методы восстановления поддерживаются в вашей СУБД. Расскажите о них.

Есть система бэкапов, работающая как через GUI так и через Command запросы





Backup and Restore

- To backup your database, type:

```
> CREATE BACKUP factbook
```

- To restore your database, type:

```
> RESTORE factbook
```

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

В этой ДБ нет ни шардинга, ни партиционирования

13. Возможно ли применить термины Data Mining, Data Warehousing и OLAP в вашей СУБД?

Определенно нет, это легковесная ДБ, в которой даже не поддерживается репликация и шардинг. Хранилища данных же предполагают терабайты данных, к которым есть доступ через какое-то API с удобной системой обеспечения сохранности данных (репликация)

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

- Разграничение прав пользователей по 4 уровням: читать, писать, создавать базы и администрировать
- XQuery Cryptographic Module (Шифрование и дешифрование данных)

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

Развитие происходит с помощью Open Source Community. В био контрибьютеров не сказано чего-либо, кроме зоны их ответственности в проекте.

Активные участники проекта: <https://basex.org/about/open-source/>

Все контрибьютеры: <https://github.com/BaseXdb/basex/graphs/contributors>

16. Создайте свои собственные данные для демонстрации работы СУБД.

В корне factbook.xml

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

Есть онлайн ресурсы, которые подробно учат использовать XQuery язык запросов:

- https://www.w3schools.com/xml/xquery_intro.asp
- <https://www.w3.org/TR/xquery-update-30/#introduction>

18. Где найти документацию и пройти обучение

Главная страница документации проекта: https://docs.basex.org/wiki/Main_Page

19. Как быть в курсе происходящего

Ссылка на ленту новостей проекта: <https://basex.org/post/>