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

XQuery

In the **XQuery** mode, XPath and XQuery expressions can be entered in the input bar.

To evaluate the following example queries, type them in the input bar and press Enter or click on the *Run query* button (green triangle) adjacent to the input bar:

Query	Description
`//country`	Return all `country` elements.
`//country[name = "Switzerland"]`	Return the `country` element of "Switzerland".

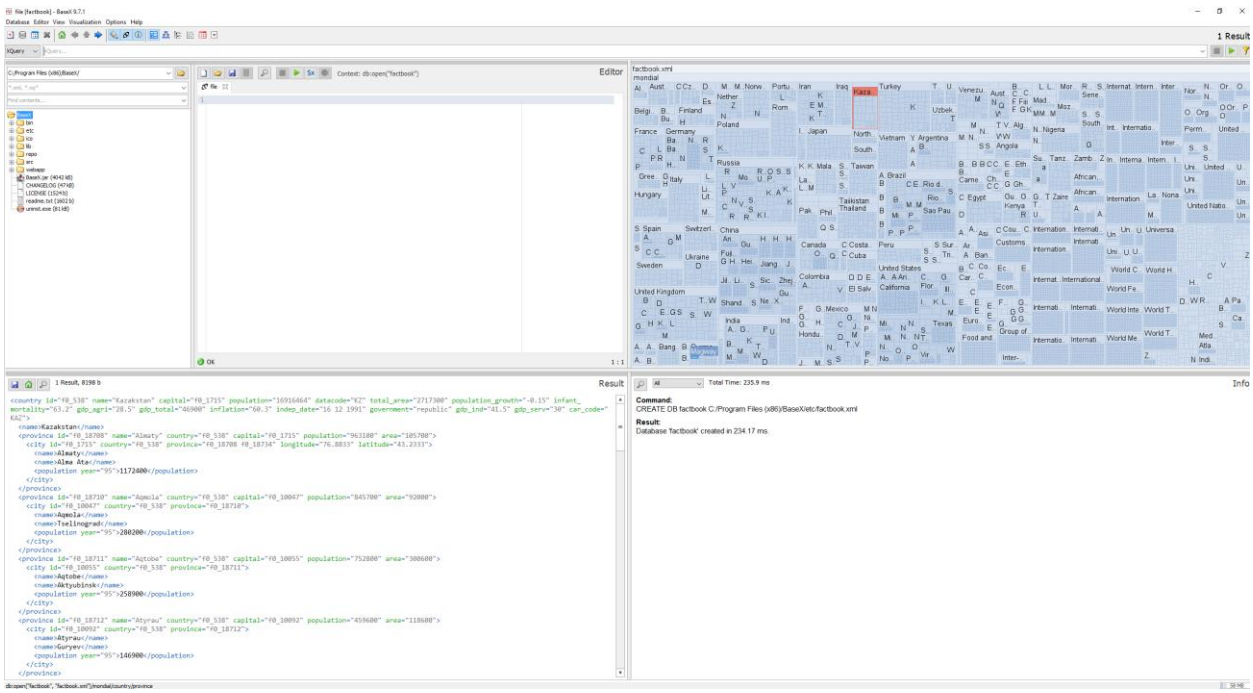
`for $city in //city`
`where $city/population > 1000000`
`order by $city ascending`
`return $city/name`
	Return the names of all cities with a population larger than one million and order the results by the name of the city.

Command

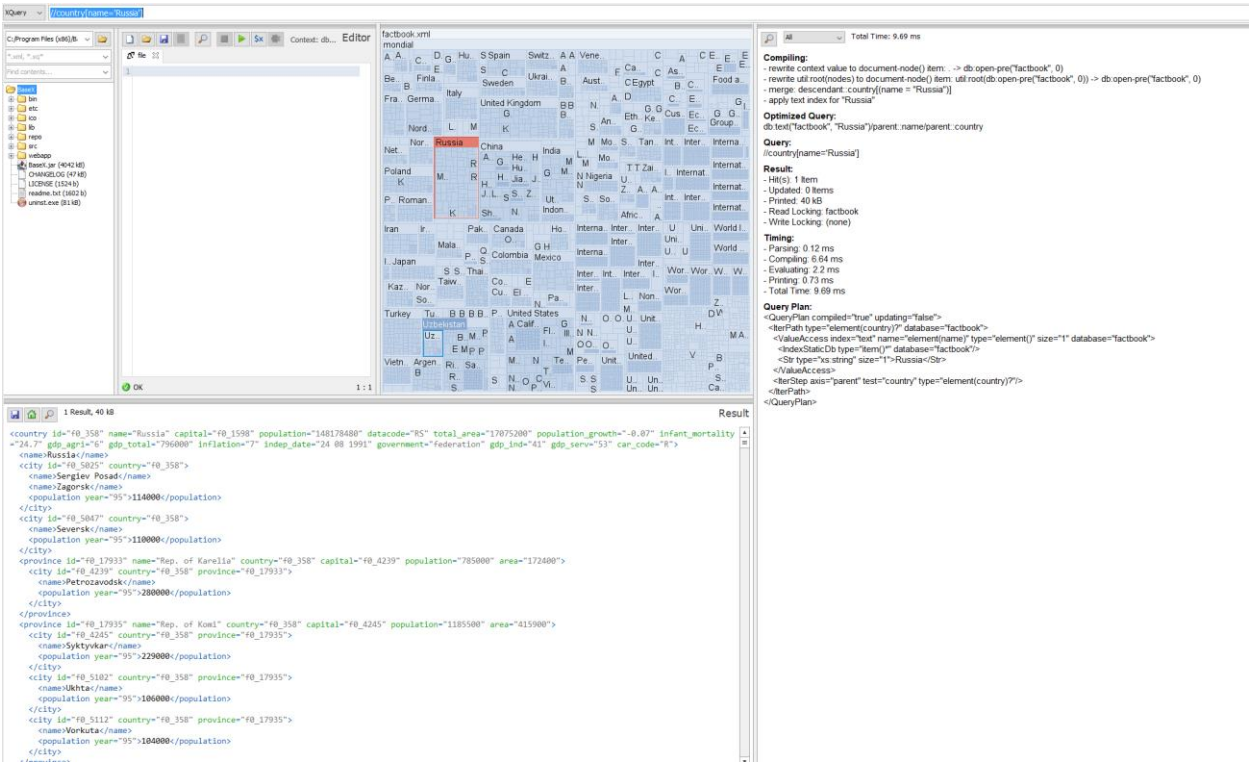
In the **Command** mode, BaseX Commands can be entered and executed. Just try the following examples:

- `INFO` : Returns system information.
- `CREATE DB TEST` : Creates an empty database named "TEST".
- `LIST` : Lists all databases.

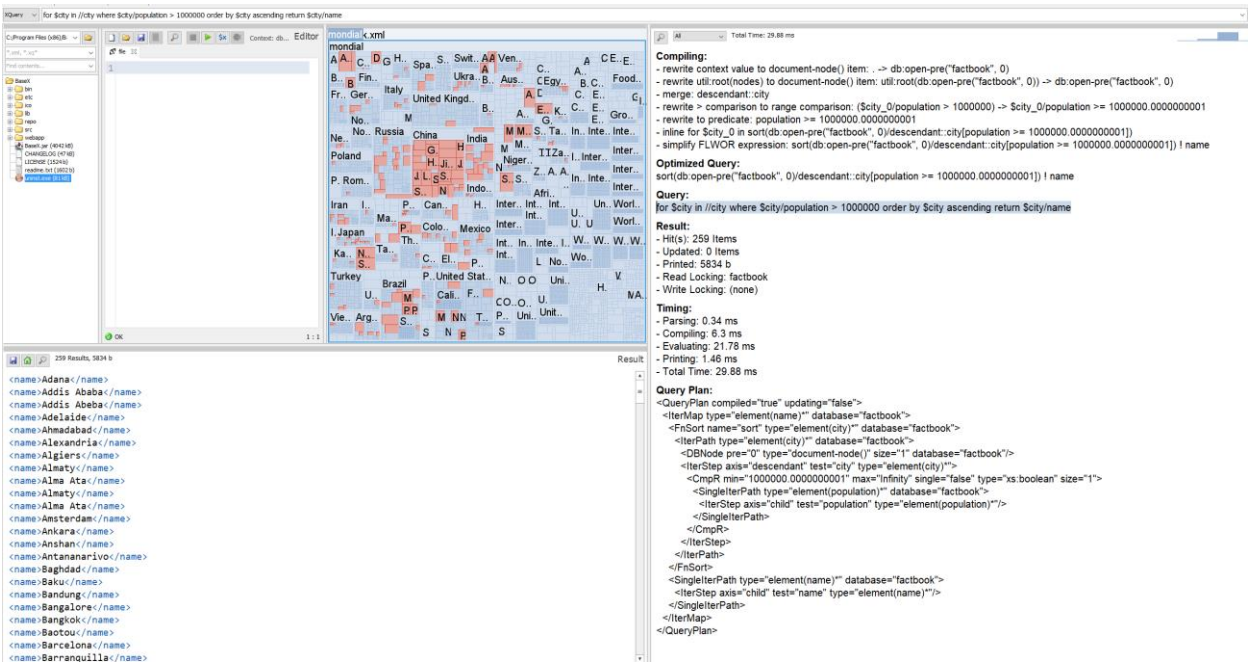
Создание БД из xml:



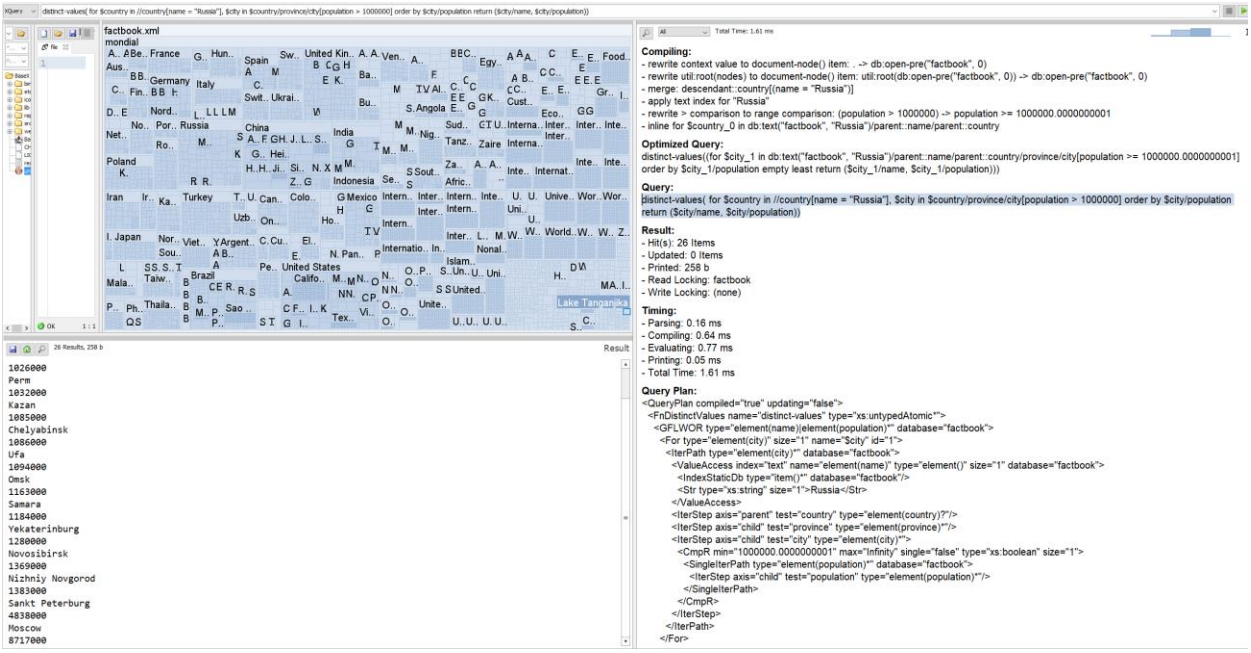
Запрашиваем информацию по России:



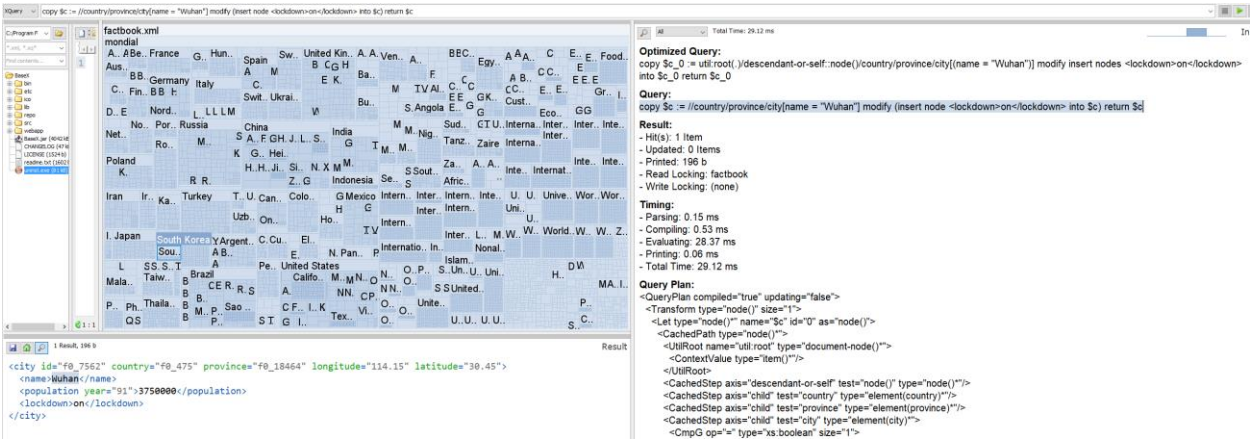
Запрашиваем информацию по городам миллионникам в лексикографическом порядке:



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)	The name index contains references to the names of all elements and attributes in a database. It contains some basic statistical information, such as the number of occurrence of a name
Path Index (system)	The path index stores all distinct paths of the documents in the database. It contains additional statistical information, such as the number of occurrence of a path, its distinct string values, and the minimum/maximum of numeric values

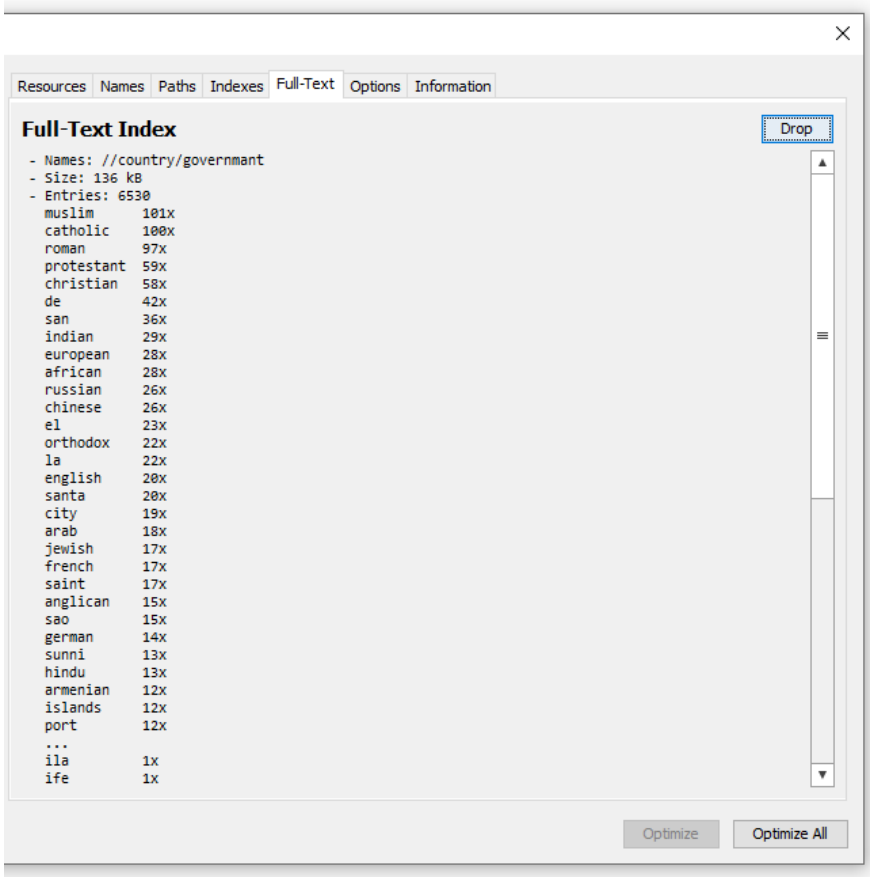
Document Index (system)	The document index contains references to all document nodes in a database.
Text Index	This index references text nodes of documents. It will be utilized to accelerate string comparisons in path expressions
Attribute Index	Similar to the text index, this index speeds up string and range comparisons on attribute values
Token Index	In many XML dialects, such as HTML or DITA, multiple tokens are stored in attribute values. The token index can be created to speed up the retrieval of these tokens
Full-Text Index	The Full-Text index contains the normalized tokens of text nodes of a document. It is utilized to speed up queries with the `contains text` expression, and it is capable of processing wildcard and fuzzy search operations
Custom Index	With XQuery, it is comparatively easy to create your own, custom index structures

Пример создания индекса по правительству в стране:

Command:
SET FTINCLUDE //country/governmant

Command:
CREATE INDEX fulltext

Result:
Index 'fulltext' created in 21.31 ms.



8. Как строится процесс выполнения запросов в вашей СУБД?

An XQuery expression is evaluated in multiple steps:

1. At parse time, the query string – an XQuery main module – is transformed to a tree representation, the *abstract syntax tree* (AST).
2. At compile time, the syntax tree is decorated with additional information (type information, expression properties); expressions are relocated, simplified, or pre-evaluated:
    1. Logical optimizations are *context-independent*. They can be applied no matter which data will be processed later on.
    2. Physical optimizations rely on context information, such as database statistics or available indexes.
3. At evaluation time, the resulting expression tree is processed.
4. The results are returned to the user. Some expression (such as simple loops) can be evaluated in iterative manner, whereas others (such as sort operations) need to be fully evaluated before the first result is available.

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="$c" id="0" as="node()">
      <CachedPath type="node()*">
        <UtilRoot name="util:root" type="document-node()*">
          <ContextValue type="item()*"/>
        </UtilRoot>
        <CachedStep axis="descendant-or-self" test="node()" type="node()*"/>
        <CachedStep axis="child" test="country" type="element(country)*"/>
        <CachedStep axis="child" test="province" type="element(province)*"/>
        <CachedStep axis="child" test="city" type="element(city)*">
          <CmpG op="=" type="xs:boolean" size="1">
            <CachedPath type="node()*">
              <CachedStep axis="child" test="name" type="element(name)*"/>
            </CachedPath>
            <Str type="xs:string" size="1">Wuhan</Str>
          </CmpG>
        </CachedStep>
      </CachedPath>
    </Let>
    <Insert type="empty-sequence()" size="0">
      <VarRef type="node()" size="1" name="$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?

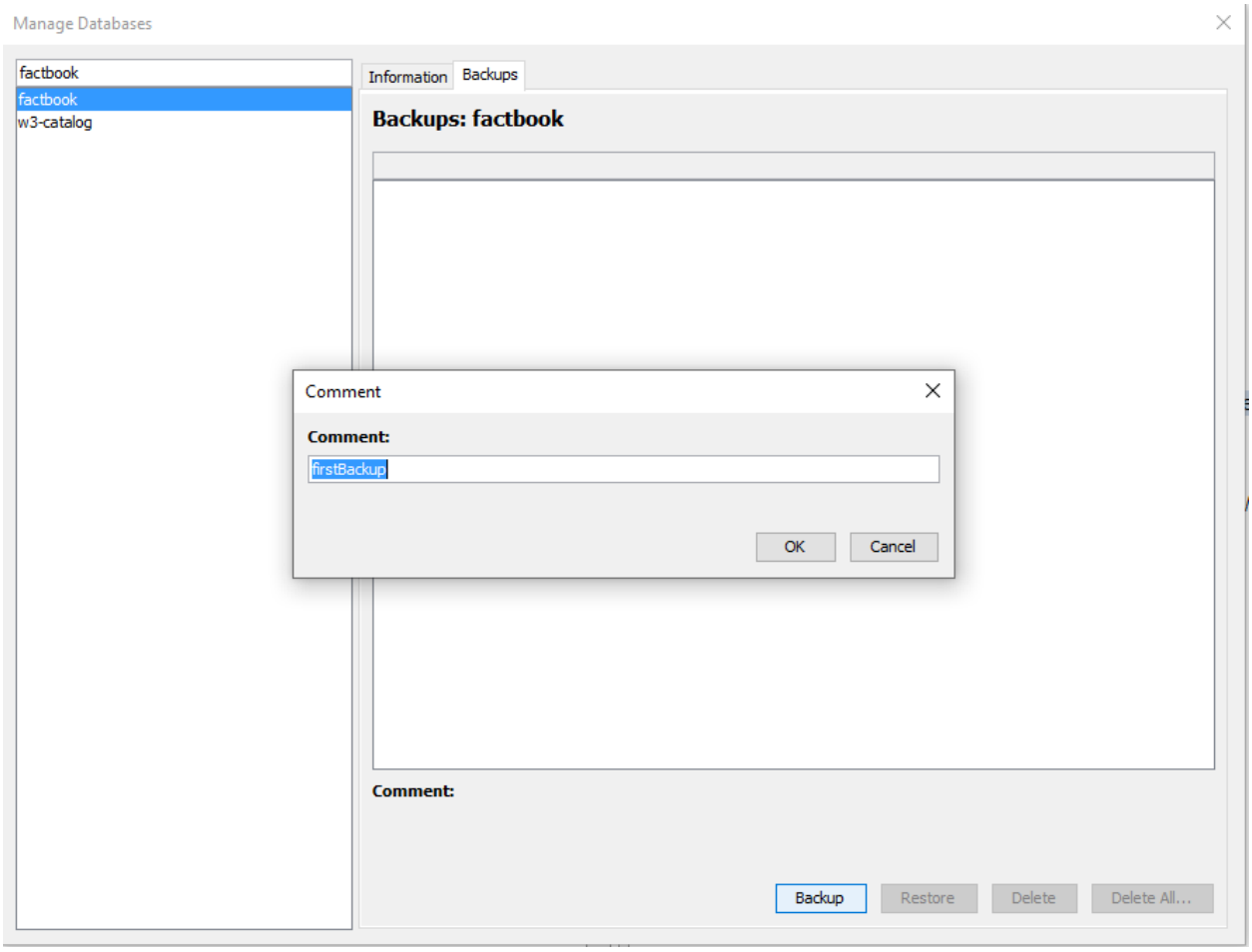
In a nutshell, a transaction is equal to a command or query. So each command or query sent to the server becomes a transaction.

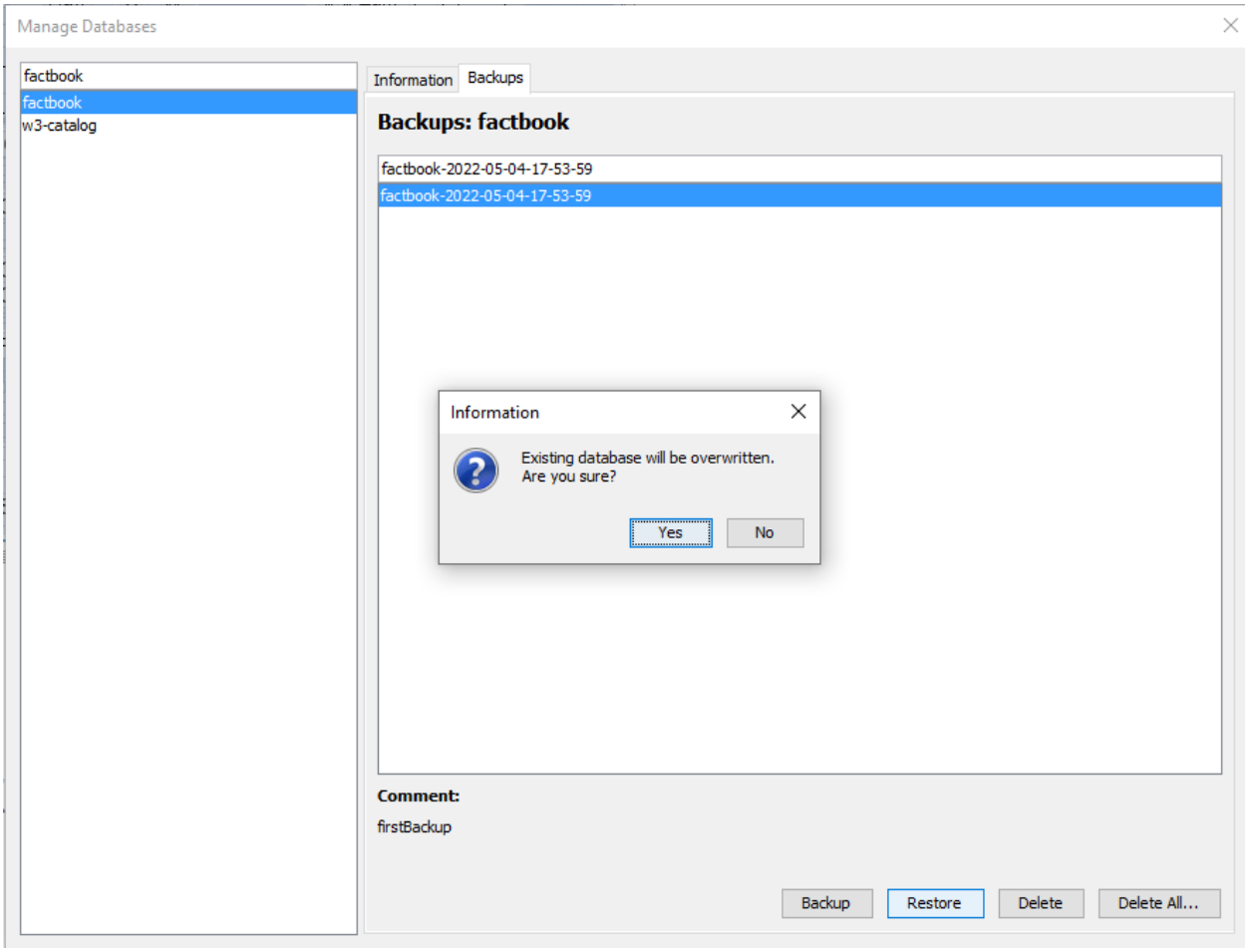
BaseX provides support for multiple read and single write operations (using preclaiming and starvation-free two phase locking). This means that:

- Read transactions are executed in parallel.
- If an updating transaction comes in, it will be queued and executed after all previous read transaction have been executed.
- Subsequent operations (read or write) will be queued until the updating transaction has completed.
- Jobs without database access will never be locked. Globally locking jobs can now be executed in parallel with non-locking jobs.
- Each database has its own queue: An update on database A will not block operations on database B. This is under the premise that it can be statically determined, i.e., before the transaction is evaluated, which databases will be accessed by a transaction
- The number of maximum parallel transactions can be adjusted with the `PARALLEL` option.
- By default, read transactions are favored, and transactions that access no databases can be evaluated even if the transactions limit has been reached. This behavior can be changed via the `FAIRLOCK` option.

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 террабайты 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.
    - Users with fine-grained authorization concept on 4 levels
    - XQuery Cryptographic Module (Encryption & Decryption of data)
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.

https://basex.org/about/open-source/

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.

https://www.w3schools.com/xml/xquery_intro.asp

https://www.w3.org/TR/xquery-update-30/#introduction

18. Где найти документацию и пройти обучение

https://docs.basex.org/wiki/Main_Page

19. Как быть в курсе происходящего

https://basex.org/post/