Database Administration

Foundational Concepts around Stardog Database Administration

Taught by:

Michael Soren
Senior Software Engineer

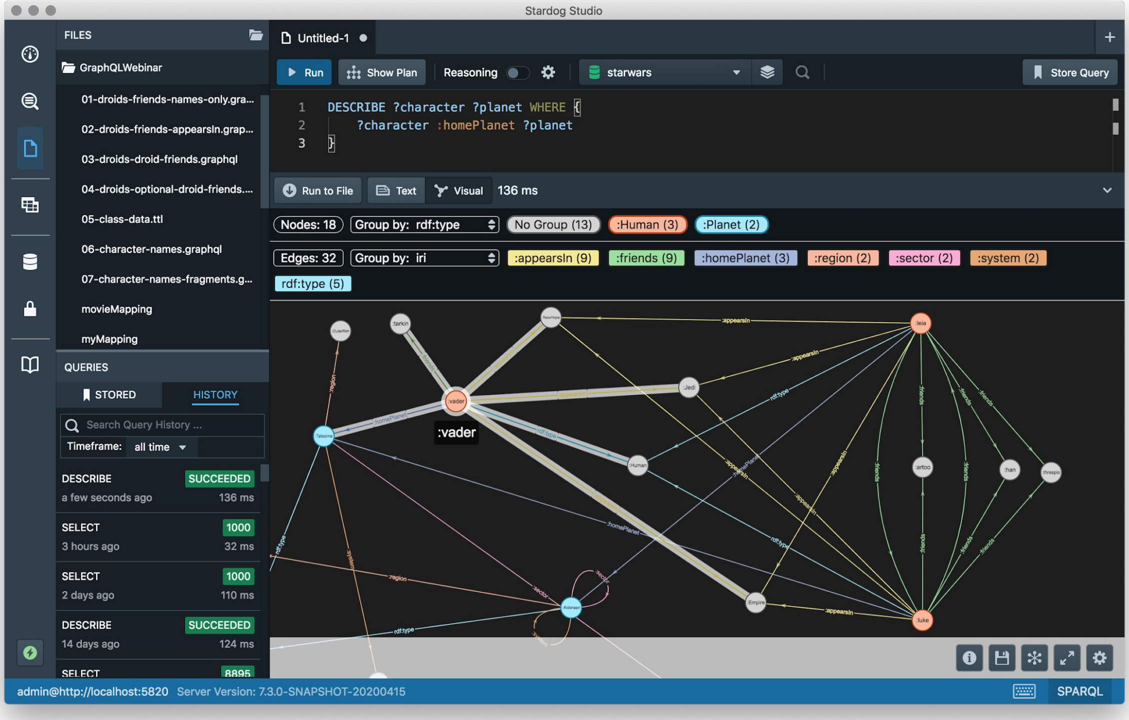
Learning Objectives

- Understand available user interface options
- Gain an understanding of basic database operations
- Configure the Stardog server as an administrator

User Interfaces

Stardog User Interface: Studio

- Studio is an easy to use graphical user interface, giving users access to most commonly used functionality
- Studio is available in browsers Firefox and Chrome at https://stardog.studio



Stardog Studio

FILES

GraphQLWebinar

- 01-droids-friends-names-only.gra...
- 02-droids-friends-appearsIn.grap...
- 03-droids-droid-friends.graphql
- 04-droids-optional-droid-friends....
- 05-class-data.ttl
- 06-character-names.graphql
- 07-character-names-fragments.g...
- movieMapping
- myMapping

QUERIES

STORED HISTORY

Search Query History ...

Timeframe: all time

DESCRIBE	SUCCEEDED
a few seconds ago	136 ms
SELECT	1000
3 hours ago	32 ms
SELECT	1000
2 days ago	110 ms
DESCRIBE	SUCCEEDED
14 days ago	124 ms
SELECT	8895

Untitled-1

Run | Show Plan | Reasoning | starwars | Store Query

```
DESCRIBE ?character ?planet WHERE {
    ?character :homePlanet ?planet
}
```

Run to File | Text | Visual | 136 ms

Nodes: 18 | Group by: rdf:type | No Group (13) | :Human (3) | :Planet (2)

Edges: 32 | Group by: iri | :appearsIn (9) | :friends (9) | :homePlanet (3) | :region (2) | :sector (2) | :system (2) | rdf:type (5)

admin@http://localhost:5820 Server Version: 7.3.0-SNAPSHOT-20200415 SPARQL

Stardog User Interface:
Command Line Interface (CLI)

- The Command Line Interface (CLI) is compact and includes the complete functionality to administer Stardog
- Due to security considerations, the CLI comes in two parts

Stardog User Interface: Command Line Interface (CLI)

stardog-admin - the administrative client:

```
usage: stardog-admin [ --server <server url> ] [ --krb5 ] [ --krb5-disable-rdns ] <command> [
<args> ]

Commands are:
    cache          Commands for working with cached views
    cluster        Commands for working with Stardog Cluster
    data-source    Commands for data source management. Data sources can be used with virtual admin
commands.
    db             Commands for working with databases
    diagnostics    Commands to help diagnose your Stardog server
    encryption     Commands that manage encryption at rest
    function       Stored function management commands
    help           Display help information
    icv            Commands for working with Stardog Integrity Constraint support
    license        License commands
    log            Commands for Stardog access and audit logs
    metadata       Modify the properties of a database
    property       Commands for viewing and modifying server properties
    query          Query management commands
    role           Commands for working with roles
    server         Commands which work with the Stardog DBMS server
    stored         Stored query management commands
    user           Commands for working with users
    version        Prints information about this version of Stardog.
    virtual        Commands for virtual graph management
    zk             Commands for working with Zookeeper Cluster

See 'stardog-admin help <command>' for more information on a specific command.
```

stardog - the user's client:

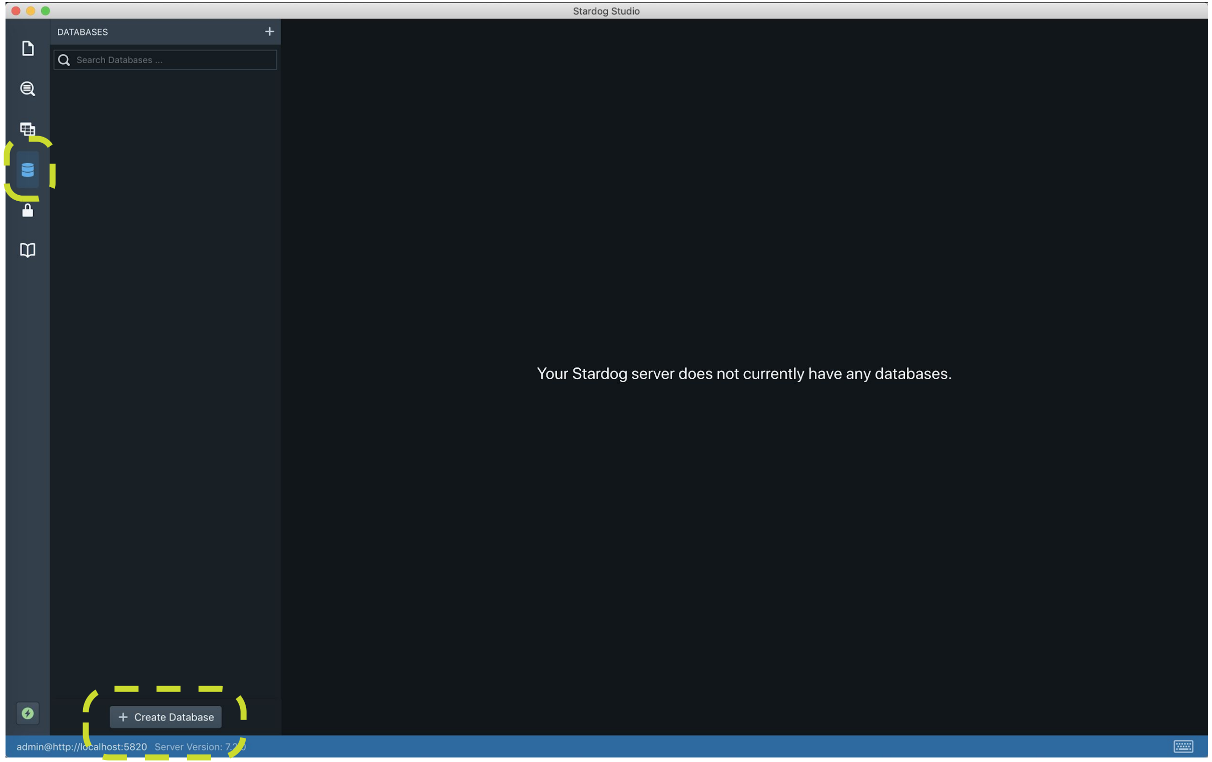
```
usage: stardog [ --krb5 ] [ --krb5-disable-rdns ] <command> [ <args> ]

Commands are:
    data           Commands which can modify or dump the contents of a database
    doc            Unstructured document processing
    file           Commands for manipulating rdf files
    graphql        Commands for working with GraphQL
    help           Display help information
    icv            Commands for working with Stardog Integrity Constraint support
    namespace      Commands which work with the namespaces defined for a database
    query          Commands which query a Stardog database
    reasoning      Commands which use the reasoning capabilities of a Stardog
database
    tx             Commands for managing transactions
    version        Prints information about this version of Stardog.

See 'stardog help <command>' for more information on a specific command.
```

Database Operations

Database Operations: Create a Database (Studio)

1. Click the Databases icon
2. Click the Create Database button
3. Type in a Database Name
4. Select a Configuration for the database
5. Click the Create button

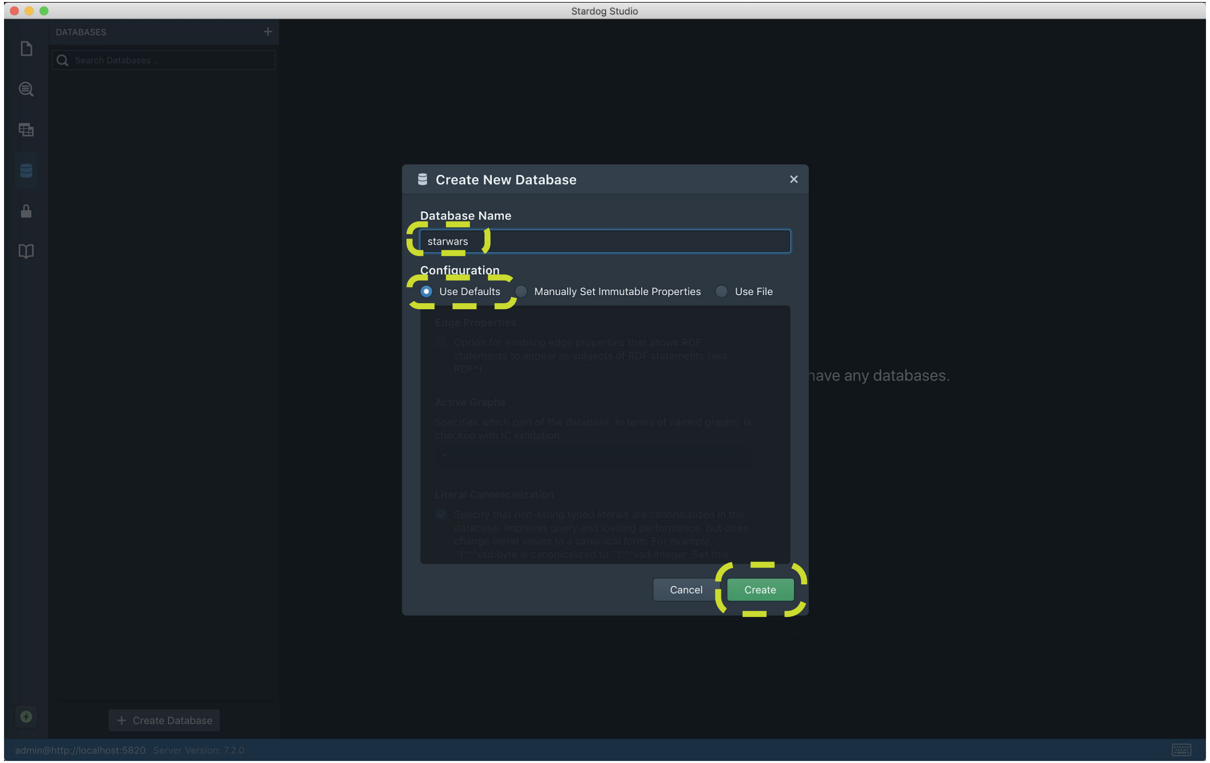
Database Operations: Create a Database (Studio)



Your Stardog server does not currently have any databases.

admin@http://localhost:5820 Server Version: 7.2.0

Database Operations:
Create a Database (Studio)



Database Name: starwars

Configuration
● Use Defaults ○ Manually Set Immutable Properties ○ Use File

Edge Properties

□ Option for enabling edge properties that allows RDF statements to appear as subjects of RDF statements (aka RDF*).

Active Graphs

Specifies which part of the database, in terms of named graphs, is checked with IC validation

*

Literal Canonicalization

☑ Specify that non-string typed literals are canonicalized in the database. Improves query and loading performance, but does change literal values to a canonical form. For example, "1"^^xsd:byte is canonicalized to "1"^^xsd:integer. Set this

Database Operations: Create a Database (CLI)

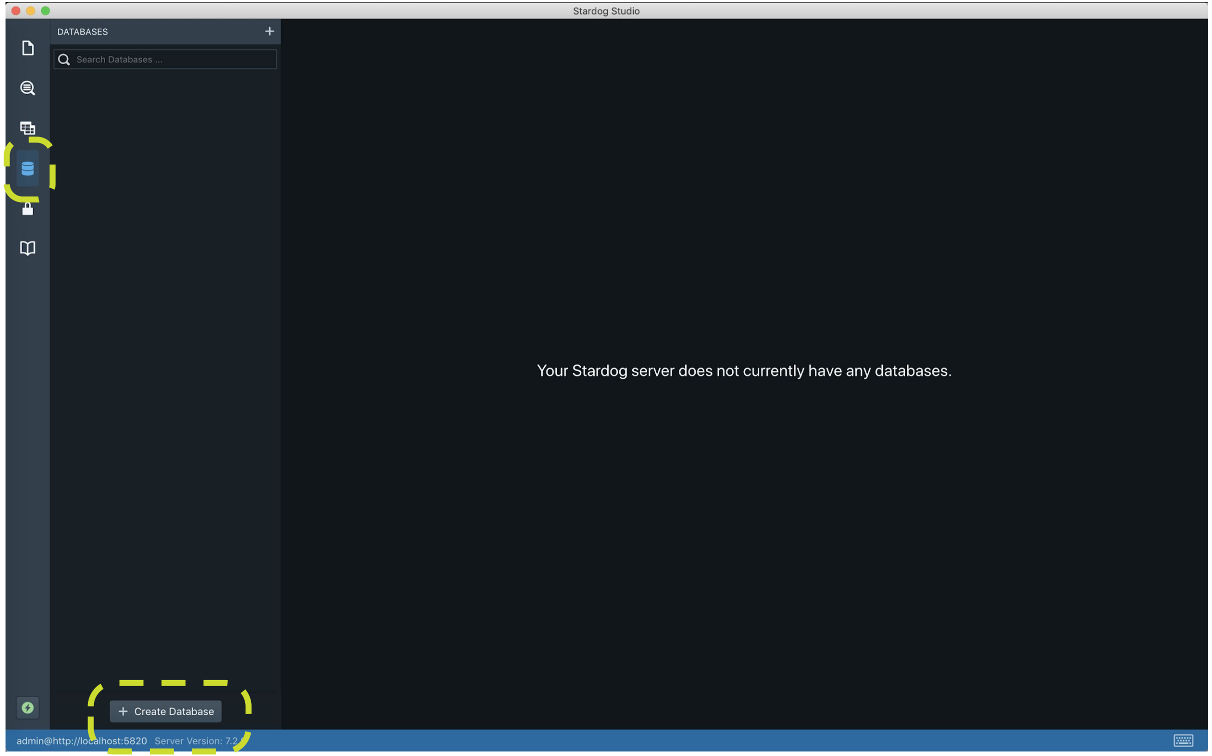
- The following command will create an empty database called "starwars":

```
$ stardog-admin db create -n starwars
```

Database Operations: Database Options (Studio)

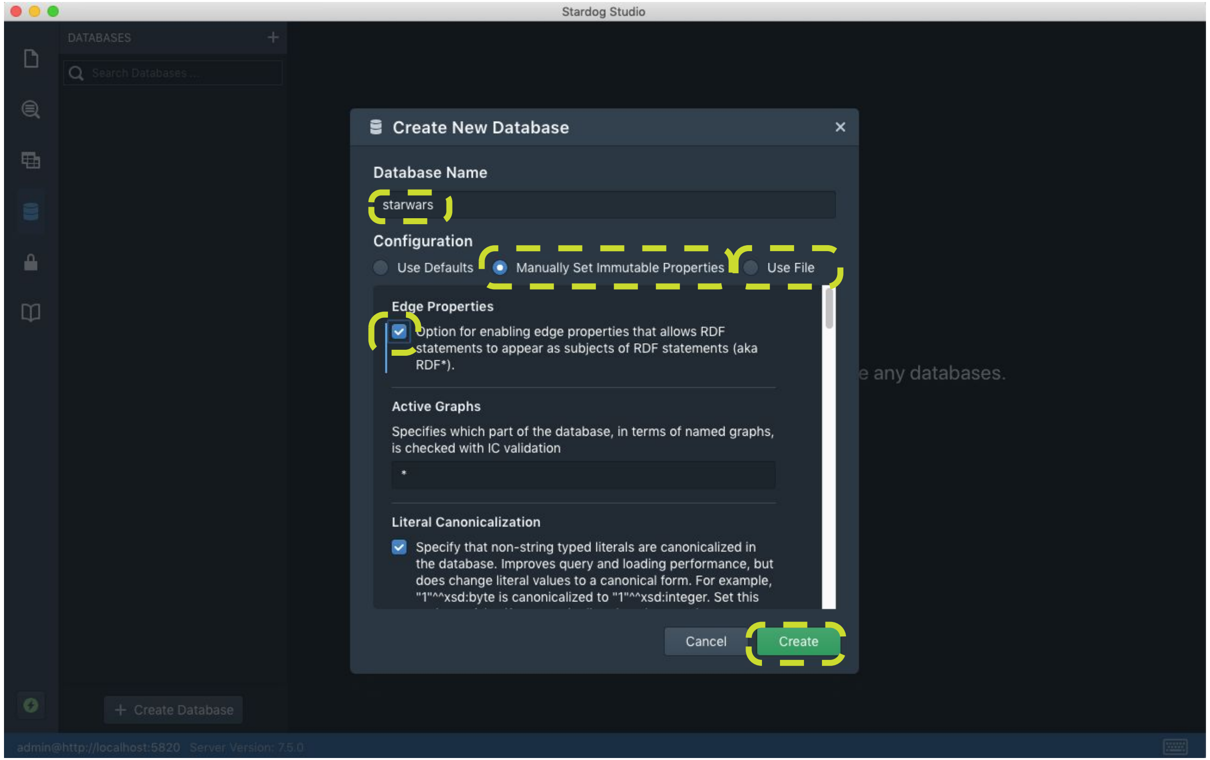
- Some database properties can only be set at creation time; click the middle radio button to manually set those immutable properties
- A properties file can also be used to configure database properties: click the "Use File" radio button on the right

Database Operations:
Database Options (Studio)



Your Stardog server does not currently have any databases.

Database Operations: Database Options (Studio)



Create New Database

Database Name

starwars

Configuration

○ Use Defaults ● Manually Set Immutable Properties ○ Use File

Edge Properties

☑ Option for enabling edge properties that allows RDF statements to appear as subjects of RDF statements (aka RDF*).

Active Graphs

Specifies which part of the database, in terms of named graphs, is checked with IC validation

*

Literal Canonicalization

☑ Specify that non-string typed literals are canonicalized in the database. Improves query and loading performance, but does change literal values to a canonical form. For example, "1"^^xsd:byte is canonicalized to "1"^^xsd:integer. Set this

Database Operations: Database Options (CLI)

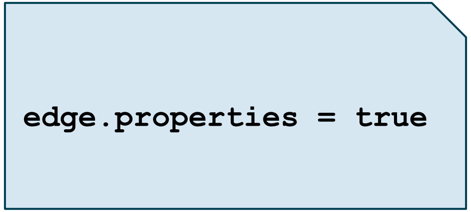
- Some database properties can only be set at creation time; set those immutable properties with the -o option
- The following command will create an empty database called “starwars” with the edge.properties option enabled:

```
$ stardog-admin db create -o edge.properties=true -n starwars
```

- A properties file can also be used to configure database properties:

```
$ stardog-admin db create -c starwars.properties -n starwars
```

starwars.properties:

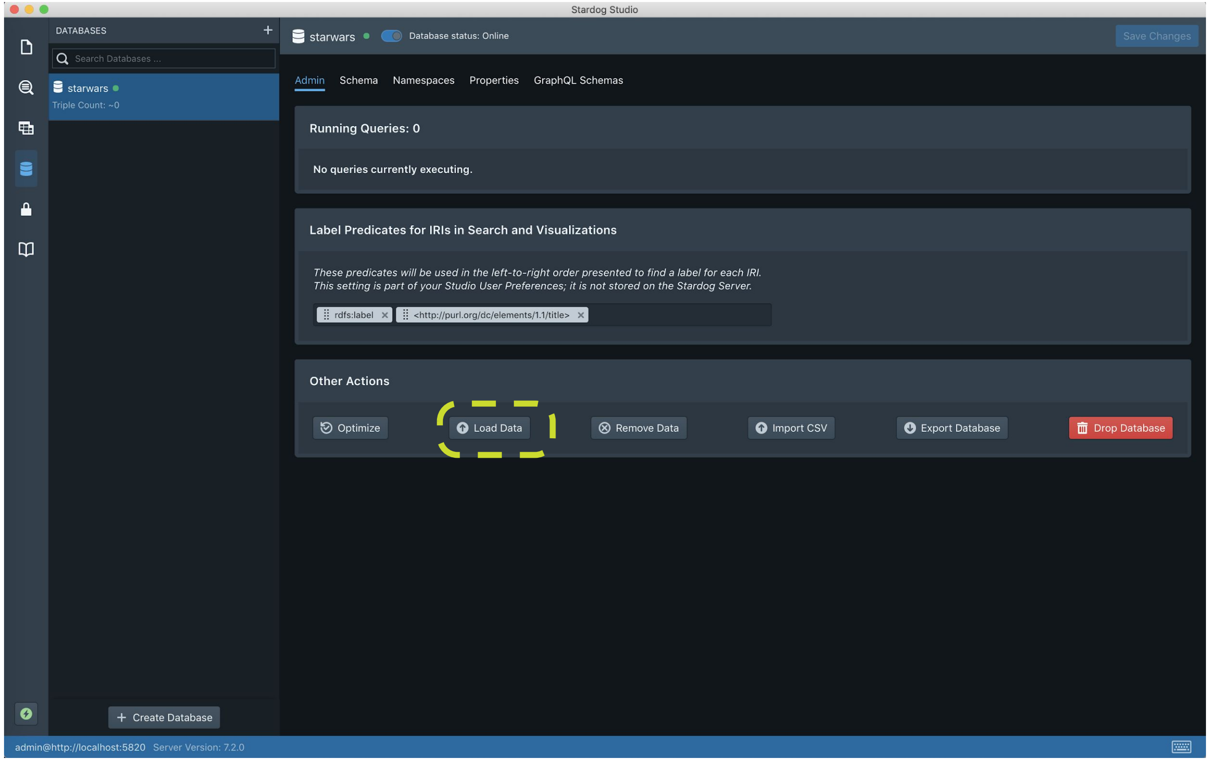


```
edge.properties = true
```

Database Operations:
Load Data into a Database (Studio)

1. Click the Load Data button
2. Click the Browse button to navigate to the file containing the data
3. Select the File Format and Compression as necessary
4. Click the Load button

Database Operations:
Load Data into a Database (Studio)



starwars

Triple Count: ~0

starwars Database status: Online

Running Queries: 0

No queries currently executing.

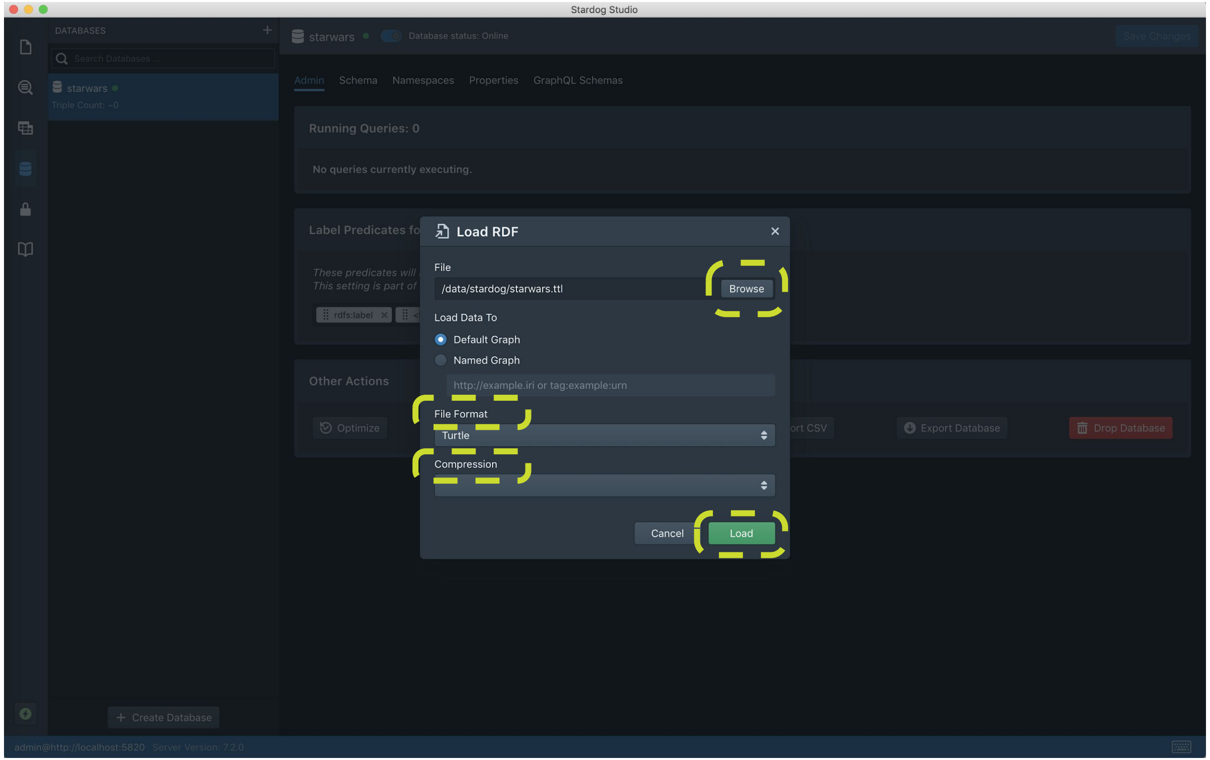
Label Predicates for IRIs in Search and Visualizations

These predicates will be used in the left-to-right order presented to find a label for each IRI.
This setting is part of your Studio User Preferences; it is not stored on the Stardog Server.

rdfs:label × <http://purl.org/dc/elements/1.1/title> ×

admin@http://localhost:5820 Server Version: 7.2.0

Database Operations:
Load Data into a Database (Studio)



Database status: Online

Running Queries: 0

No queries currently executing.

These predicates will
This setting is part of

File
/data/stardog/starwars.ttl

Load Data To
● Default Graph
○ Named Graph
http://example.iri or tag:example:urn

File Format
Turtle

Compression

admin@http://localhost:5820 Server Version: 7.2.0

Database Operations:
Load Data into a Database (CLI)

- The following command adds the data in file `/data/stardog/starwarsData.ttl` to the `starwars` database:

```
$ stardog-admin data add starwars /data/stardog/starwarsData.ttl
```

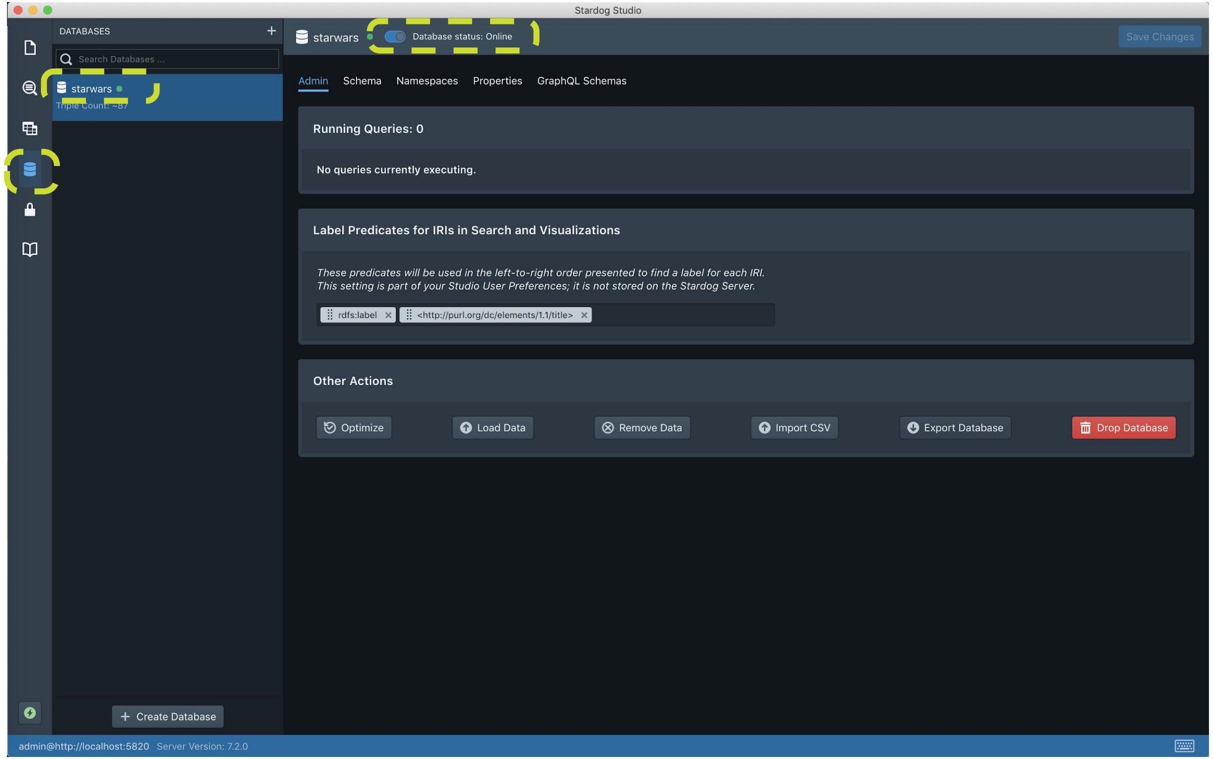
- For better performance, data can be loaded at database creation time:

```
$ stardog-admin db create -n starwars
/data/stardog/starwarsData.ttl
```

Database Operations: View/Change Status (Studio)

- A database's status must be set to Offline before certain (mutable) database properties can be changed
  1. Click the Databases icon
  2. Click the desired database
  3. Move the Database status switch to set the database offline or online

Database Operations: View/Change Status (Studio)



Running Queries: 0

No queries currently executing.

Label Predicates for IRIs in Search and Visualizations

These predicates will be used in the left-to-right order presented to find a label for each IRI.
This setting is part of your Studio User Preferences; it is not stored on the Stardog Server.

rdfs:label ✕ <http://purl.org/dc/elements/1.1/title> ✕

Other Actions

Database Operations: View/Change Status (CLI)

- The following command shows the status of the **`starwars`** database:

```
$ stardog-admin db status starwars
Database             : starwars
Status               : Online
Approx. size         : 87 triples
Queries              : None running
Open Connections     : 0
Open Transactions    : 0
Query Avg. Time      : 0.08 s
Query Rate           : 0.00 queries/sec
Plans Cached         : 3
Plan Cache Hit Ratio : 33.33%
```

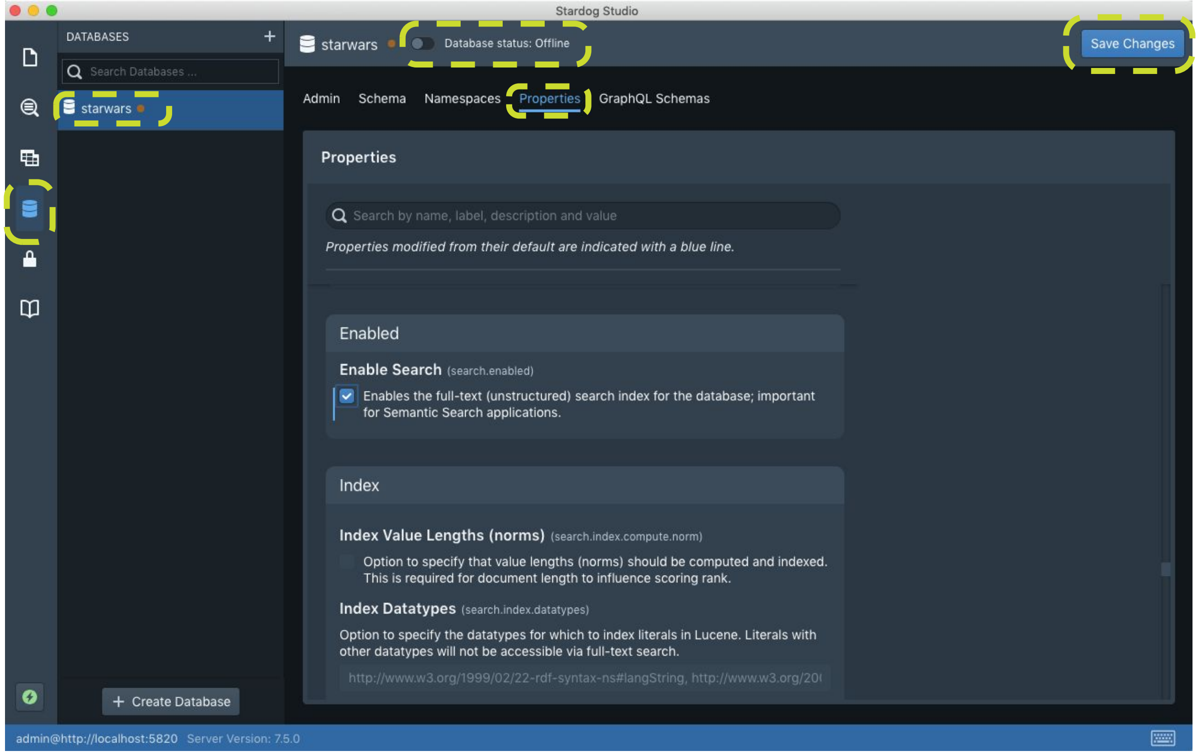
- The following command changes the status of the **`starwars`** database to **`offline`**:

```
$ stardog-admin db offline starwars
The database starwars is now offline.
```

Database Operations: View/Change Properties (Studio)

1. Click the Databases button
2. Click the database whose properties you wish to view or change
3. Click the Properties button
4. Scroll to view property settings
5. Some properties can only be set while the database is offline. Slide the Database status to set it offline
6. If desired, make property changes, then click Save Changes
7. If the database was set offline, set it back online

Database Operations: View/Change Properties (Studio)



Properties

Search by name, label, description and value

Properties modified from their default are indicated with a blue line.

Enabled

Enable Search (search.enabled)

☑ Enables the full-text (unstructured) search index for the database; important for Semantic Search applications.

Index

Index Value Lengths (norms) (search.index.compute.norm)

□ Option to specify that value lengths (norms) should be computed and indexed. This is required for document length to influence scoring rank.

Index Datatypes (search.index.datatypes)

Option to specify the datatypes for which to index literals in Lucene. Literals with other datatypes will not be accessible via full-text search.

http://www.w3.org/1999/02/22-rdf-syntax-ns#langString, http://www.w3.org/20(

admin@http://localhost:5820 Server Version: 7.5.0

Database Operations: View/Change Properties (CLI)

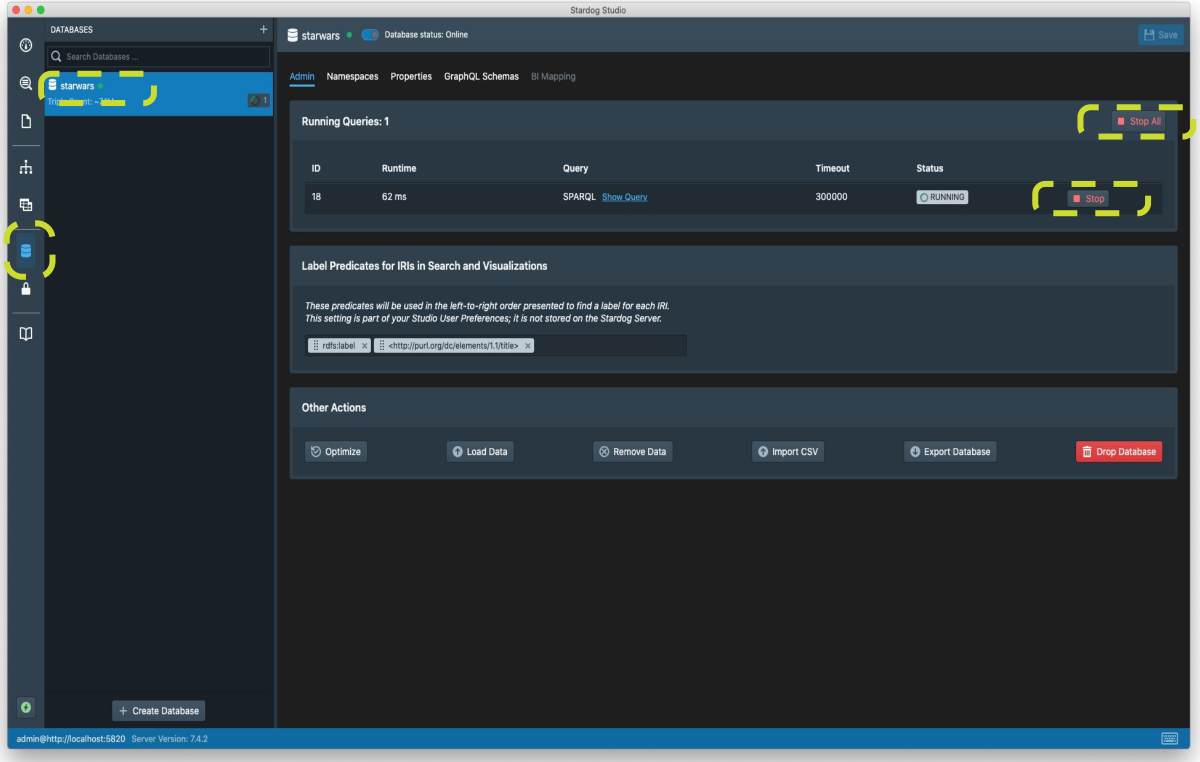
- The following commands get properties and set the specified property for the `starwars` database:

```
$ stardog-admin metadata get starwars
+-----------------------------+-----------------------------------------------------------------------------------+
|                   Option    |                                      Value                                        |
+-----------------------------+-----------------------------------------------------------------------------------+
| database.archetypes         |                                                                                   |
| database.connection.timeout | 10m                                                                               |
| database.creator            | admin                                                                             |
| database.name               | starwars                                                                          |
|                                              .                                                                   |
|                                              .                                                                   |
|                                              .                                                                   |

$ stardog-admin metadata get -o search.enabled starwars
+----------------+-------+
|     Option     | Value |
+----------------+-------+
| search.enabled | false |
+----------------+-------+

$ stardog-admin metadata set -o search.enabled=true -- starwars
Cannot change configuration option while the database starwars is online: search.enabled=false
$ stardog-admin db status offline starwars
The database starwars is now offline.
$ stardog-admin metadata set -o search.enabled=true -- starwars
The option(s) for the database 'starwars' were successfully set.
$ stardog-admin db status online starwars
The database starwars is now online.
```

Database Operations: View/Stop Running Queries (Studio)

1. Click the Databases icon
2. Click the desired database
3. Click the Stop button to kill the associated running query, or the Stop All button to stop all queries

Database Operations: View/Stop Running Queries (Studio)



Running Queries: 1

ID	Runtime	Query	Timeout	Status	
18	62 ms	SPARQL Show Query	300000	RUNNING	Stop

Label Predicates for IRIs in Search and Visualizations

These predicates will be used in the left-to-right order presented to find a label for each IRI.
This setting is part of your Studio User Preferences; it is not stored on the Stardog Server.

rdfs:label × <http://purl.org/dc/elements/1.1/title> ×

Other Actions

Optimize | Load Data | Remove Data | Import CSV | Export Database | Drop Database

admin@http://localhost:5820 Server Version: 7.4.2

Database Operations: View/Stop Running Queries (CLI)

- The following command lists running queries:

```
$ stardog-admin query list
+----------+----------+-------+------------+--------------+---------+--------------+
| Query ID | Database | User  | Query Type | Elapsed time | Status  |   Timeout    |
+----------+----------+-------+------------+--------------+---------+--------------+
| 35       | starwars | admin | SPARQL     | 00:00:11.789 | Running | 00:05:00.000 |
+----------+----------+-------+------------+--------------+---------+--------------+
```

- The following command stops a running query:

```
$ stardog-admin query kill 35
Query 35 killed successfully.
```

Database Operations: Backing Up a Database (CLI)

- Backup creates a physical copy of a database and its metadata
- Can be performed while the database is online; other activity continues, but data updates after the process begins will not be in the backup
- By default, backups are stored in $STARDOG_HOME/`.backup;` Change this location by setting the `backup.dir` server property in the [$STARDOG_HOME/stardog.properties]() file
- To specify a location outside of $STARDOG_HOME, use the `--to` option or set the `backup.location` server property in the [$STARDOG_HOME/stardog.properties]() file
- Users can perform a backup using only the Command Line Interface (CLI):

```
$ stardog-admin db backup <database-name>
```

For example, if the environmental variable STARDOG_HOME is set to `/stardog_home`,

```
$ stardog-admin db backup starwars
Backup of starwars save to /stardog_home/.backup/starwars/2020-11-30 in 00:00:00.070
```

Database Operations: Restoring a Database (CLI)

- Restores a database from a backup; create a new database from a backup using the `-n` option
- The location of the backup should be the full path to the backup, not the location of the backup directory
- Users can perform a restore using only the Command Line Interface (CLI):

```
$ stardog-admin db restore -n <new-database name> <backup-location>
```

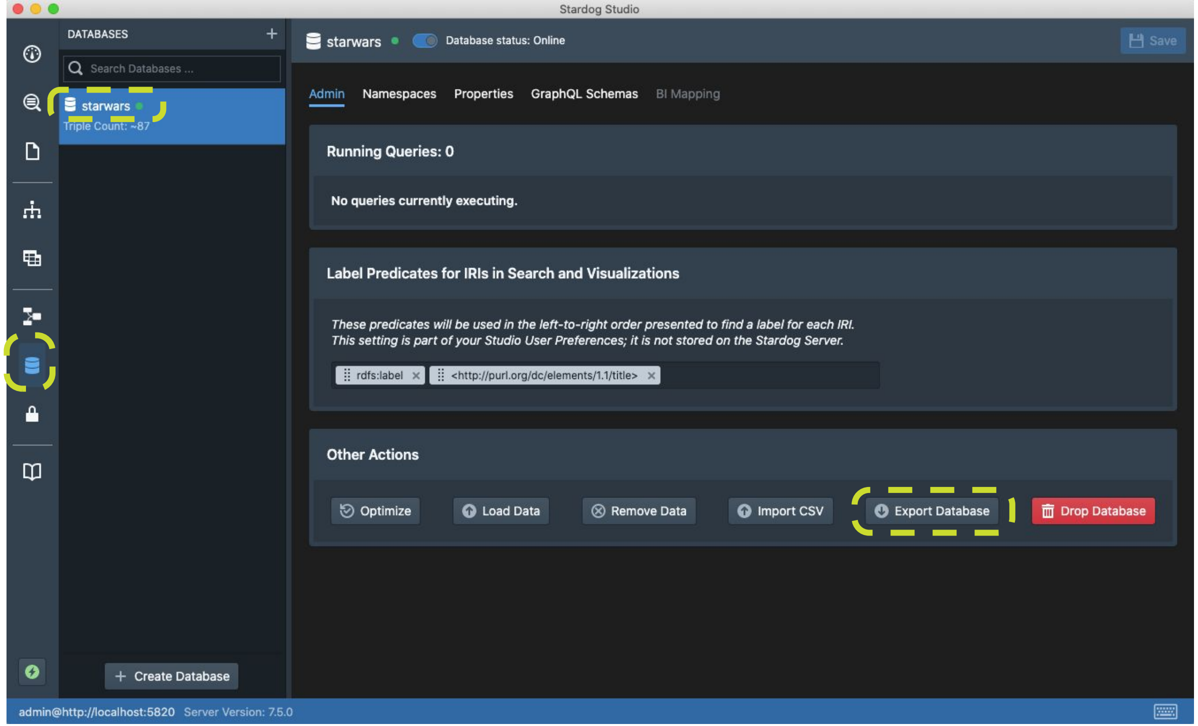
From the previous example:

```
$ stardog-admin db restore -n starwars2 /stardog_home/.backup/starwars/2020-11-30
Node 472fb379-40c6-466d-bf91-517623c1bbbc: Successfully restored database 'starwars2' from
Backup(/stardog_home/.backup/starwars/2020-11-30) in 00:00:00.448.
```

Database Operations: Export a Database (Studio)

1. Click the Databases icon
2. Click the desired database
3. Click the Export Database button
4. Select the File Format to be exported in the drop down
5. Click the Export button

Database Operations: Export a Database (Studio)



Running Queries: 0

No queries currently executing.

Label Predicates for IRIs in Search and Visualizations

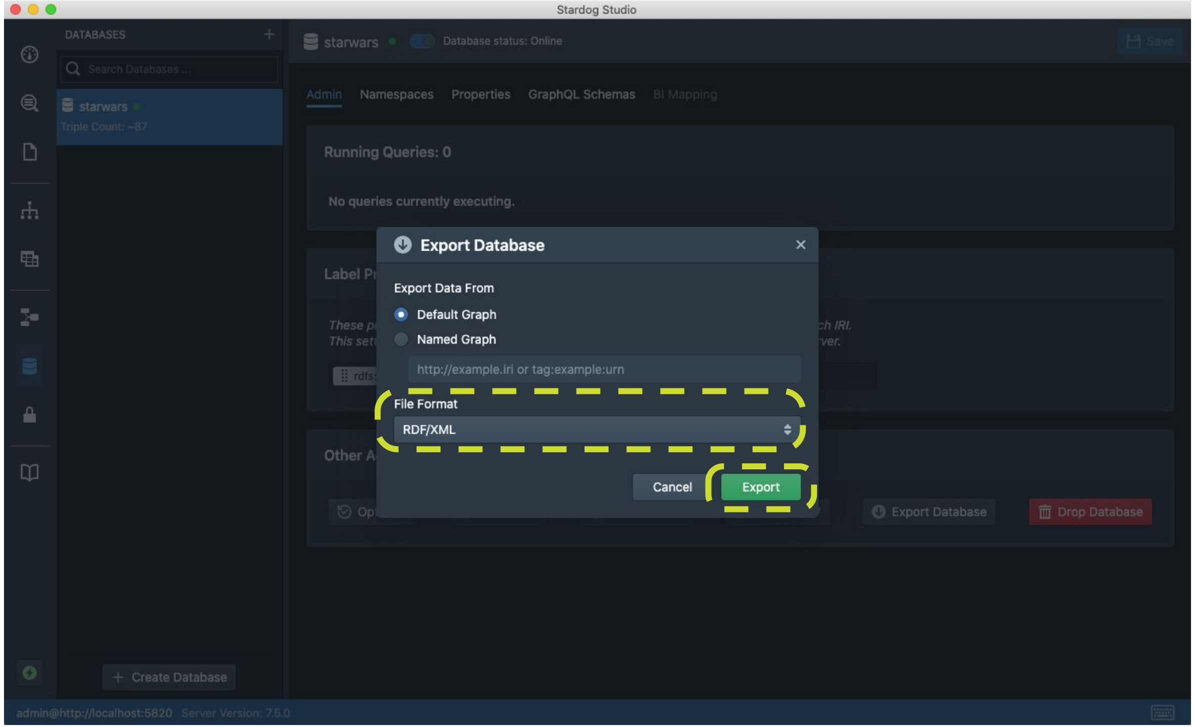
These predicates will be used in the left-to-right order presented to find a label for each IRI. This setting is part of your Studio User Preferences; it is not stored on the Stardog Server.

rdfs:label × <http://purl.org/dc/elements/1.1/title> ×

Other Actions

admin@http://localhost:5820 Server Version: 7.5.0

Database Operations:
Export a Database (Studio)



Stardog Studio

starwars · Database status: Online

Running Queries: 0

No queries currently executing.

Export Database

Export Data From

● Default Graph

○ Named Graph

http://example.iri or tag:example:urn

File Format: RDF/XML

admin@http://localhost:5820 Server Version: 7.5.0

Database Operations: Export a Database (CLI)

- The following command saves the contents of the starwars database into a standard RDF file:

```
$ stardog data export --format RDF/XML <database-name>
<export-file>
```

- Using the starwars database as an example:

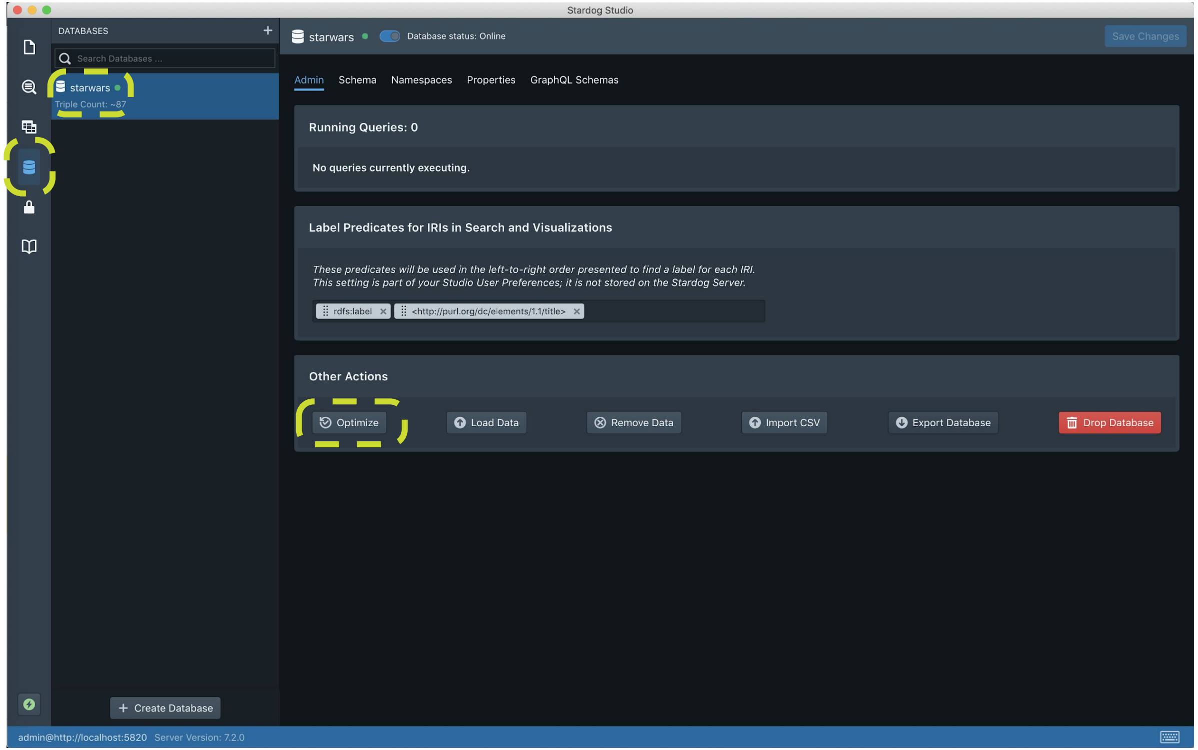
```
$ stardog data export --format RDF/XML starwars starwars.rdf
```

Database Operations: Optimize a Database (Studio)

- Optimizes the indexes of a database after the database has been heavily modified
- Stardog server must be running

1. Click the Databases icon
2. Click the desired database
3. Click the Optimize button

Database Operations: Optimize a Database (Studio)



Stardog Studio

DATABASES

Search Databases ...

starwars
Triple Count: ~87

Create Database

starwars ● Database status: Online

Save Changes

Admin | Schema | Namespaces | Properties | GraphQL Schemas

Running Queries: 0

No queries currently executing.

Label Predicates for IRIs in Search and Visualizations

These predicates will be used in the left-to-right order presented to find a label for each IRI.
This setting is part of your Studio User Preferences; it is not stored on the Stardog Server.

rdfs:label × | <http://purl.org/dc/elements/1.1/title> ×

Other Actions

Optimize | Load Data | Remove Data | Import CSV | Export Database | Drop Database

admin@http://localhost:5820 Server Version: 7.2.0

Database Operations: Optimize a Database (CLI)

- Optimizes the indexes of a database after the database has been heavily modified
- Stardog server must be running

```
$ stardog-admin db optimize starwars
starwars was successfully optimized.
```

Database Operations:
Repair a Database (CLI)

- Attempts to repair a corrupted database
- Prerequisites:
  - Stardog server must be running
  - The database must be offline
- Can be performed only from the Command Line Interface (CLI):

```
$ stardog-admin db repair <database-name>
```

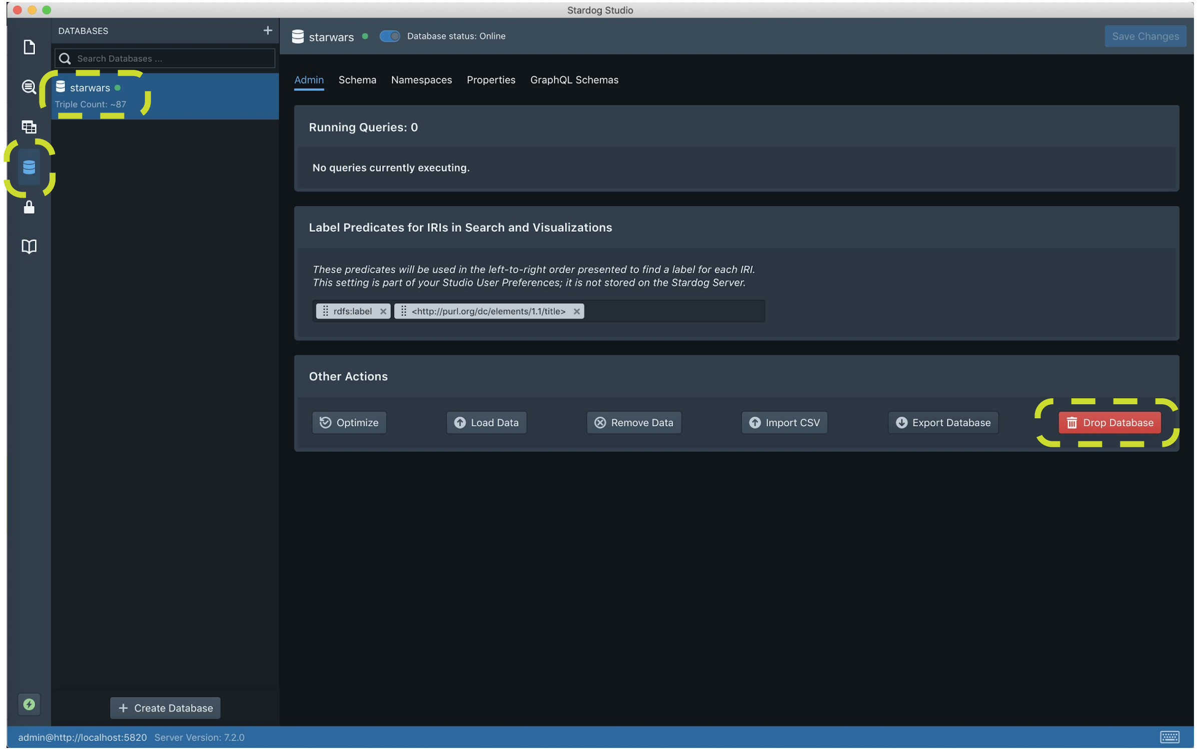
For example:

```
$ stardog-admin db repair starwars
```

Database Operations: Drop a Database (Studio)

1. Click the Databases icon
2. Click the desired database
3. Click the Drop Database button

Database Operations: Drop a Database (Studio)



Running Queries: 0

No queries currently executing.

Label Predicates for IRIs in Search and Visualizations

These predicates will be used in the left-to-right order presented to find a label for each IRI.
This setting is part of your Studio User Preferences; it is not stored on the Stardog Server.

rdfs:label ✕ <http://purl.org/dc/elements/1.1/title> ✕

Other Actions

Database Operations: Drop a Database (CLI)

- The following command drops a database:

  ```
  $ stardog-admin db drop <database-name>
  ```

  For example:

  ```
  $ stardog-admin db drop starwars
  ```

  ```
  Successfully deleted database 'starwars'.
  ```

Demo

Server Administration

Stardog Server Administration

- Configuring Server Memory - Two Ways:
  - Memory Configuration I: Setting JVM memory parameters
  - Memory Configuration II: Setting server property `memory.mode`
- Logging
  - Logging Basics
  - Enabling Logging
  - Log File Rotation
- Stardog server running as a linux service

Stardog Server Administration: Memory Configuration I

- Memory available to Stardog is often the most important factor in performance
- Stardog uses both JVM heap memory and direct (native) operating system memory
- By default, Stardog sets maximum JVM memory to 2GB, direct memory to 1GB; good for small databases that have <100 million triples
- For larger databases, use the table in the Stardog documentation to set these two values appropriately
- To change these values from the default, set and export the environment variable `STARDOG_SERVER_JAVA_ARGS` with the standard JVM options. For example, to set the JVM heap memory to 3GB and the direct memory to 4GB:

```
$ export STARDOG_SERVER_JAVA_ARGS="-Xmx3g -Xms3g -XX:MaxDirectMemorySize=4g"
```

Stardog Server Administration: Memory Configuration II

- Stardog defines four memory consumption modes with the `memory.mode` server property

`default`	Unknown, or equal amounts of queries and updates
`read_optimized`	Infrequent/small transactions
`write_optimized`	Optimal loading and update performance
`bulk_load`	Bulk loading very large databases; when complete, this memory configuration should be changed

- Set the server property `memory.mode` in the [$STARDOG_HOME/stardog.properties]() file
- Server must be restarted for changes to take effect

Stardog Server Administration: Logging Basics

- Three types of logs:
  - Access log: All connection-based events, such as queries and transactional changes, and authentication events. Connection boundary events are not logged.
  - Audit log: All events; a superset of the access log.
  - Slow query log: All queries whose execution exceeds the slow query time property `logging.slow_query.time` (by default, 10 seconds).
- All logs are disabled by default.
- Logs can be written in type `text` (to be read by humans) or `binary` (machine readable).
- By default, logging is directed to the `$STARDOG_HOME` directory with the file name:
  - Access: `access.log`
  - Audit: `audit.log`
  - Slow Query: `slow_query.log`

Stardog Server Administration: Enabling Logging

- To enable Access logging, add the following to the $STARDOG_HOME/stardog.properties file:

```
logging.access.enabled = true
logging.access.type = text
```

- To enable Audit logging, add the following to the $STARDOG_HOME/stardog.properties file:

```
logging.audit.enabled = true
logging.audit.type = text
```

- To enable Slow Query logging, add the following to the $STARDOG_HOME/stardog.properties file:

```
logging.slow_query.enabled = true
logging.slow_query.type = text
```

- Server must be restarted for changes to take effect

Stardog Server Administration: Log File Rotation

- Log entries can be split into multiple files by defining a file rotation strategy
- The trigger to start a new file can be:
  - The size of a file (in bytes):

    ```
    logging.XXX.rotation.type = size
    logging.XXX.rotation.limit = 1000000
    ```

  - A fixed time interval:

    ```
    logging.XXX.rotation.type = time
    logging.XXX.rotation.limit = 1d
    ```

where *`XXX`* is one of `access, audit,` or `slow_query` for that log

Stardog Server Running as a Linux Service

- The Stardog server can be run as a service on a linux system using `systemd`:
  - The operating system will start the Stardog server when it starts up
  - The operating system will shut down the Stardog server when it shuts down
  - The operating system will restart the Stardog server in the event of abnormal termination
- The Stardog server’s environment can be customized by defining environment variables in the file `/etc/stardog.env.sh.`

For example, if the file `/etc/stardog.env.sh` has contents:

```
export STARDOG_HOME=/var/opt/stardog
```

This will cause the `STARDOG_HOME` environment variable to be set to `/var/opt/stardog.`

All databases and other files will be stored in this directory

Learning Objectives

Learning Objectives

- Understand available user interface options
- Gain an understanding of basic database operations
- Configure the Stardog server as an administrator

Thank you