



# The FHIR SQL Builder

2024-05-06

*The FHIR SQL Builder*

InterSystems IRIS Data Platform 2024-05-06

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, and TrakCare are registered trademarks of InterSystems Corporation. HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, and InterSystems TotalView™ For Asset Management are trademarks of InterSystems Corporation. TrakCare is a registered trademark in Australia and the European Union.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

**InterSystems Worldwide Response Center (WRC)**

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: [support@InterSystems.com](mailto:support@InterSystems.com)

# Table of Contents

<b>The FHIR SQL Builder</b> .....	<b>1</b>
1 Schema Generation Overview .....	2
2 Configuration .....	2
3 Analyze the FHIR Repository .....	2
4 Creating a New Transformation Specification .....	4
4.1 Adding a Primitive Element to the Schema .....	4
4.2 Adding Data from Collections to the Schema .....	5
4.3 Adding an Object to the Schema .....	6
4.4 Viewing Transformation Specifications .....	7
5 Exporting, Importing, and Copying a Transformation Specification .....	7
5.1 Exporting a Transformation Specification .....	8
5.2 Importing a Transformation Specification .....	8
5.3 Copying a Transformation Specification .....	8
6 Create the Projection .....	8
7 Next Steps: Tune Query Performance .....	10

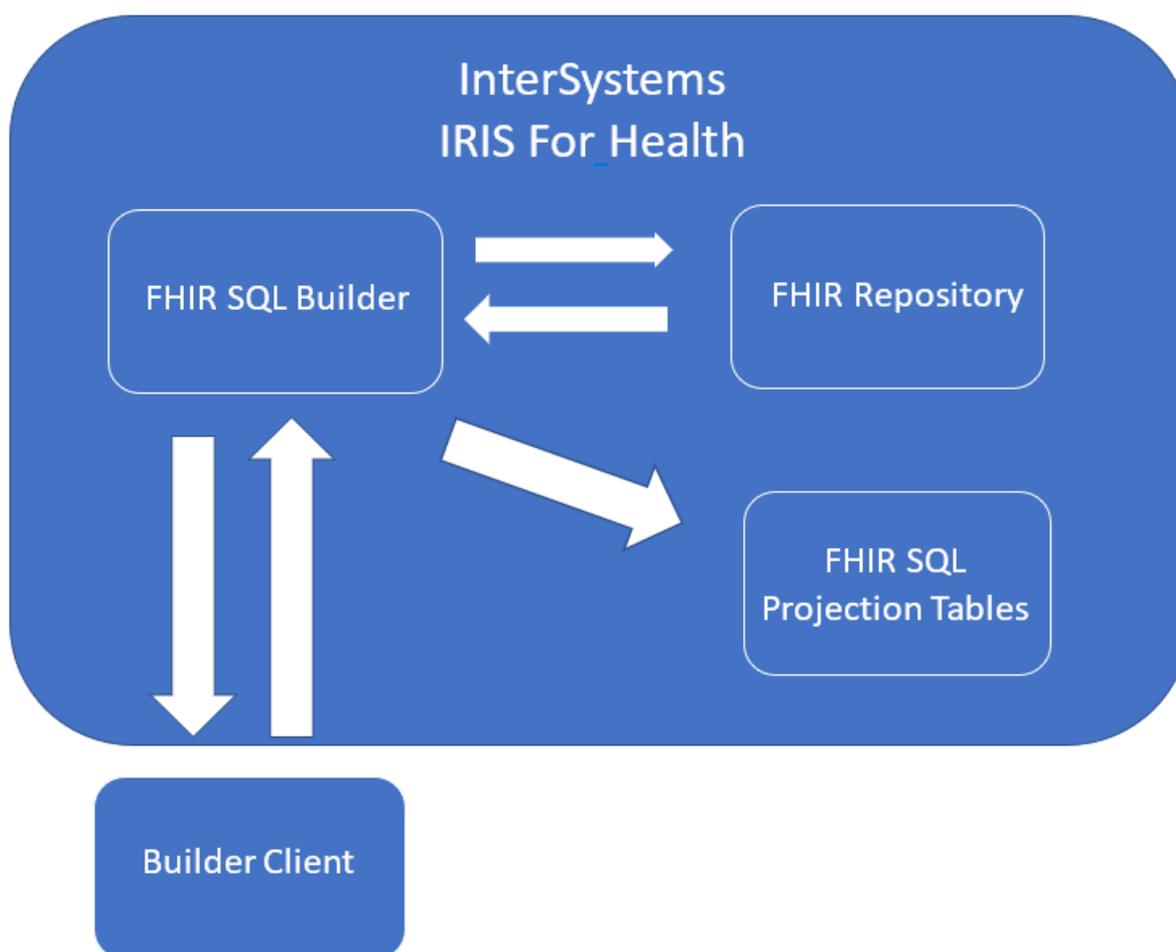


# The FHIR SQL Builder

The FHIR® SQL Builder, or *Builder*, is a component of InterSystems IRIS for Health. It is a sophisticated projection tool used to create custom SQL schemas using data in an InterSystems IRIS for Health FHIR repository without moving the data to a separate SQL repository. The Builder is designed specifically to work with FHIR repositories and multi-model databases in InterSystems IRIS for Health.

The objective of the Builder is to enable data analysts and business intelligence developers to work with FHIR using familiar analytic tools, without having to learn a new query syntax. FHIR data is encoded in a complex directed graph and cannot be queried using standard SQL syntax. A graph-based query language, FHIRPath, is designed to query FHIR data, but it is non-relational. Enabling a data steward to create a customized SQL projection of their FHIR repository, using tables, columns, and indexes, the Builder makes it possible for data analysts to query FHIR data without the complexity of learning FHIRPath or the FHIR search syntax.

The following diagram shows the relationships between the Builder and other components in InterSystems IRIS for Health.



The Builder analyzes the InterSystems IRIS for Health FHIR repository to generate summary information, including the types of resources, elements, and values it contains, as well as the number of each type of resource. You decide which FHIR resources and elements to include in your custom SQL projection and how to map them.

# 1 Schema Generation Overview

To generate a schema the main actions are:

1. Analyze the FHIR repository.
2. Present the analysis to the user.
3. Use decisions made by the user to create the required tables.

The analysis process needs to examine enough of the repository to provide useful information, while limiting how much it examines to conserve time and resources. You can make decisions that influence this balance when configuring the analysis task. You can also configure the task to run at a time when there is less demand for computing resources.

## 2 Configuration

If your license key is in place before the installation of InterSystems IRIS for Health finishes, the Web Applications that implement the Builder will already be enabled. Otherwise, you will need to manually enable them. On the InterSystems IRIS for Health instance that is running the Builder:

- In the Management Portal, go to **System Administration > Security > Applications > Web Applications**.
- On the **Web Applications** page, enable the applications `/csp/fhirsq` and `/csp/fhirsq/api/ui` by selecting the name of the application, which opens the **Edit Web Application** page.
- On the **General** tab of the **Web Application** page, click the **Enable Application** check box and then click **Save**.

The FHIR SQL Builder is designed to work with FHIR repositories across different instances of InterSystems IRIS for Health. On each instance of InterSystems IRIS for Health that runs a FHIR server, follow the steps above to enable `/csp/fhirsq/api/repository`. Only one instance should have the `/csp/fhirsq/api/ui` API enabled, but any number of instances can have the `/csp/fhirsq/api/repository` API enabled.

Once the applications have been properly enabled, you will be able to open the Builder at `http://hostname:portnumber/csp/fhirsq/index.html#/` on versions 2023.2 and later or `http://hostname:portnumber/csp/fhirsq/index.csp#/` on version 2023.1, where `hostname` is the name of the host of your InterSystems IRIS for Health instance (this can be `localhost`) and `portnumber` is the port number of your InterSystems IRIS for Health instance. Users that access the Builder will need to have the `%Admin_Manage` resource, or else will encounter a 403 error. For more information about giving certain users access to this resource, see [“Manage Roles”](#) in [Roles](#).

Starting the FHIR SQL Builder application brings you to the home page. The work area is divided into sections which let you configure the **Analyses**, **Transformation Specifications**, and **Projections** of your repository.

## 3 Analyze the FHIR Repository

The goal of analyzing your FHIR Repository is to summarize the available structural relationships, constituent elements, and embedded collections of the repository. It forms the basis for defining a transformation specification by outlining how various elements are related to one another and providing a range of values you can expect for a given element. The analysis task does not need to look at every record in the repository; is up to you to decide the size of the sample that you will analyze.

To configure an analysis, click **New** on the right side of the **Analyses** section to open the **New FHIR Analysis** dialog. Fill out this dialog as follows:

- **FHIR repository** – You can select a repository from the drop-down list or create a new repository by clicking **New** to the right of the field to open the **New FHIR Repository Configuration** dialog. Fill out this dialog as follows:
  - **Name** – Enter a name for the repository.
  - **Host** – Enter the DNS name or IP address of the host where the repository you wish to analyze is hosted. If the Builder UI and the repository are on the same instance, you can use `localhost`.
  - **Port** – Enter the port used to access the repository.
  - **SSL Configuration** (optional) – If you are using SSL, select the SSL configuration you want to use. For information about creating a new SSL configuration, refer to “[Create or Edit a TLS Configuration](#)” in [About Configurations](#).
  - **Credentials** – Select credentials from the drop-down list or create new credentials by clicking the **New** button to the right of the field to open the **New Credentials** dialog. Fill out this dialog as follows:
    - **Name** – Enter a name for this credentials object.
    - **Username** – Enter the username. This must be the name of a User on the current instance of InterSystems IRIS for Health. To see a full list of the available names, open the Management Portal and navigate to **System Administration > Security Management > Users**.
    - **Password** – Enter the password.
    - Click **Save** to return to the **New FHIR Repository Configuration** dialog.
  - **FHIR Repository URL** – Select a repository from the drop-down list. When the values entered for **Name**, **Host**, **Port**, and **Credentials** establish a valid connection, this field provides a list of available FHIR repositories.
  - Click **Save** to return to the **New FHIR Analysis** dialog.
- **Selectivity Percentage** – Limits analysis to a percentage of the FHIR repository. If used, **Maximum Records** cannot be used.
- **Maximum Records** – Limits analysis to a maximum number of records in the FHIR repository. If used, **Selectivity Percentage** cannot be used.
- **Defer Start of Task** – Select this check box to run the analysis task at a later point in time. If selected, enter a **Start Date** and a **Start Time**.
- Click **Launch Analysis Task** to start analyzing the FHIR repository.

**Important:** When creating a new pair of credentials, be sure that the User has the Roles it needs to access the FHIR Repository. These roles are: `%HS_DB_{namespace}`, `%DB_{namespace}`, and the FHIR repository databases (for example, `%DB_DEMOX0001R` and `%DB_DEMOX0001V`). If the User does not have these roles, nothing will appear in the **FHIR Repository URL** drop-down and you will be unable to select a repository for your analysis.

You will see the newly started analysis on the list of analyses on the Builder home page. Columns provide the following information:

- **FHIR Repository** – The name you provided when configuring the analysis.
- **Start Time** – The date and time the analysis started.
- **Last Modified** – The date and time you last modified the analysis.
- **Status** – The possible values are: **Running**, **Stopping**, **Stopped**, **Completed**, and **Errored**.

- **Total Resources** – The number of FHIR records analyzed.
- **Percent Complete** – The amount of the analysis completed as a percentage of the anticipated total.
- **Actions** – Provides buttons you can use to control the progress of the analysis.
  -  **Resume** – Resumes running a paused analysis.
  -  **Pause** – Pauses a running analysis.
  -  **Delete** – Deletes the analysis of the repository. An analysis cannot be deleted if a Transformation Specification depends on it.

## 4 Creating a New Transformation Specification

Once the FHIR Repository analysis is complete, you can use the resulting information to create a transformation specification. A transformation specification structures the table schemas it generates and determines which resources and elements should be included in the projection. Transformation specifications can also be exported, imported, and copied; see [Exporting, Importing, and Copying a Transformation Specification](#) for more information.

To create a new Transformation Specification, click the **New** button on the right side of the **Transformation Specifications** section to open the **New Transformation Specification** dialog. Fill out this dialog as follows:

- **Name** – Provide a unique name for this specification.
- **Analysis** – Select a FHIR Repository analysis from the drop-down list. Analyses are identified by the repository name and the date the analysis was created.
- **Description (Optional)** — Provide a brief description of the transformation specification.
- Click **Create Transformation Specification** to open the **Edit Transformation Specification** page.

The page opens listing resources and elements in the FHIR repository and their counts. Selecting a resource, such as **Patient**, shows the elements in that resource.

The structure presented by the analysis reflects the often deeply nested structure of the FHIR repository. Elements are:

- primitives, such as string, boolean, date, or number
- objects, which contain properties that are primitives, objects, or collections
- collections, which contain primitives, objects, or collections

Resources are structured like trees, and selecting a node (which represents either an object or a collection in the repository) opens the properties it contains below it. Leafs (which represent primitive elements) across the repository may have identical names, like “code.” As such, it is best to refer to a primitive element in the tree by the full path through the nodes from the parent resource to the child leaf (for example, “AllergyIntolerance.code.coding.code” as opposed to “Patient.code.coding.code”).

### 4.1 Adding a Primitive Element to the Schema

Select a primitive element to open a panel that allows you to add the element to the schema. For primitives, such as strings or numbers, you can click **Show Histogram** to view a histogram of the unique values in the repository. You can edit the name of the column that will appear in the table by editing the **Column name** field. Select the check box label **Index** if you

would like to add an index on this column in the table. For information regarding which elements you may want to add an index to, see “[What to Index](#)” in [Optimizing Query Performance](#).

Finally, click **Add To Projection** to add the element to the Projection.

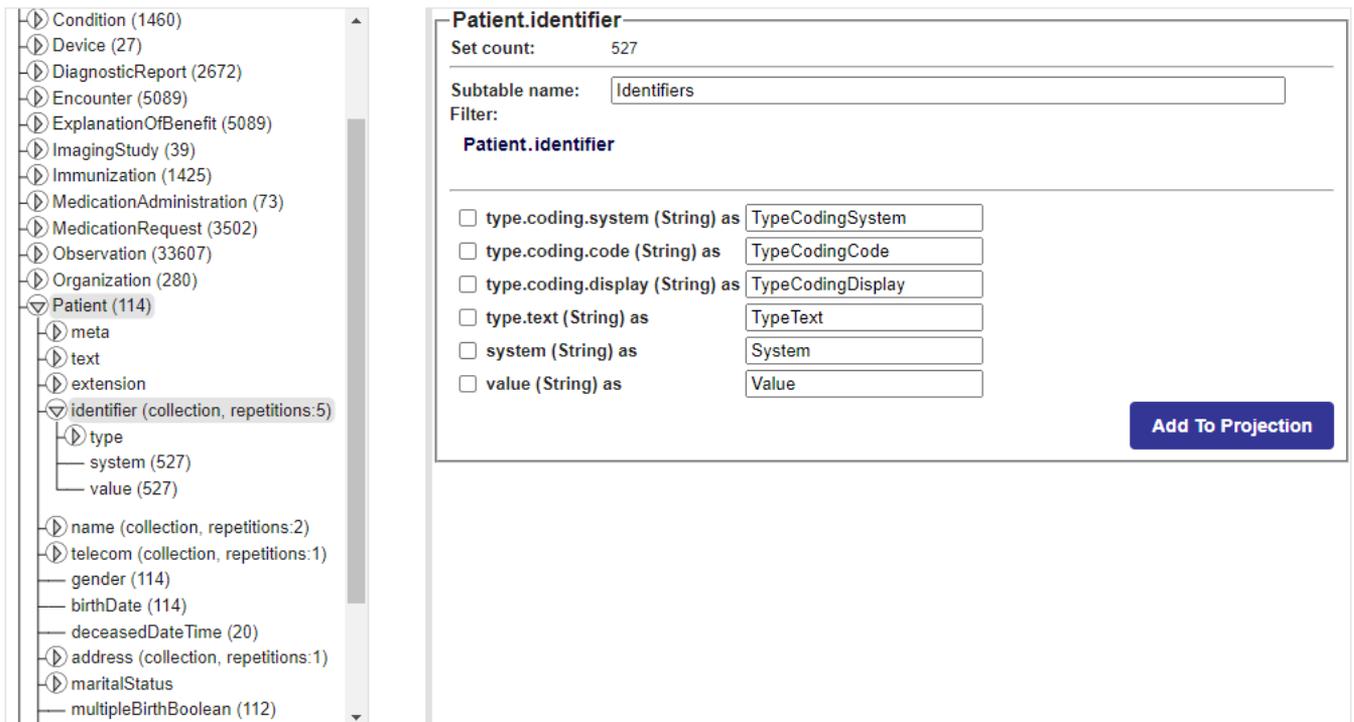
The **Currently Selected Items** table on the edit page shows the current state of the schema. It has columns the following columns:

- **Table** – The name of the table the added column is for. You can add elements from multiple different resources to a schema in a Transformation Specification; these elements will appear in the tables as specified by the values in this column.
- **Column** – The name of the column that will appear in the table.
- **Type** – The datatype of the contents of the column.
- **Index** – Indicated whether an index will be created for the column.
- **Actions** – Provides buttons you can use to edit or delete the specific element.
  -  **Edit** – Opens the panel that you used to add the element to the schema, enabling you to edit your preferences for this column, including setting filters, in the schema.
  -  **Delete** – Deletes the column specified by the from the schema.

You can add as many elements to the schema as needed. Click **Done** to return to the home page, where you will find the new specification listed.

## 4.2 Adding Data from Collections to the Schema

Some elements have collections associated with them. Selecting a collection will bring up a selection panel where all the elements at the bottom node of the collection are listed in rows. An example, using the **Identifier** collection of a **Patient**, is below. Note that this particular collection has both a object and primitive elements within it.



The screenshot shows the FHIR SQL Builder interface. On the left is a tree view of resources. The 'Patient' resource (114) is expanded, showing its sub-resources: 'meta', 'text', 'extension', 'identifier (collection, repetitions:5)', 'name (collection, repetitions:2)', 'telecom (collection, repetitions:1)', 'gender (114)', 'birthDate (114)', 'deceasedDateTime (20)', 'address (collection, repetitions:1)', 'maritalStatus', and 'multipleBirthBoolean (112)'. The 'identifier' collection is selected, showing its sub-elements: 'type', 'system (527)', and 'value (527)'. On the right is the configuration panel for 'Patient.identifier'. It shows 'Set count: 527' and a 'Subtable name' field containing 'Identifiers'. Below is a 'Filter:' section with the title 'Patient.identifier'. There are six rows of checkboxes and text input fields: 'type.coding.system (String) as TypeCodingSystem', 'type.coding.code (String) as TypeCodingCode', 'type.coding.display (String) as TypeCodingDisplay', 'type.text (String) as TypeText', 'system (String) as System', and 'value (String) as Value'. A blue 'Add To Projection' button is located at the bottom right of the configuration panel.

Click the check boxes next to the primitive elements you would like to add to the projection. When adding data from a collection in this manner, the Builder will automatically create a subtable for this data and it will not be included in the table of the parent resource. The name of the table is identified by the **Subtable name** field, which you can edit. The rows of this table correspond to individual entries of the collection and the columns are the elements of the collection you have specified, plus some special columns that help identify what resource they came from and where in the collection they appear.

However, it is sometimes desirable to have data from a collection stored within a table representing the resource it came from and not in a subtable. To do so, you should navigate to a primitive element you would like to add by clicking through the collection and selecting **Add to Projection**. However, this action will automatically populate the column with the value of the first element of the collection. As this element is likely arbitrary, it is not recommended to leave the column as such. Instead, you should use the **Filter** option.

The **Filter** option allows you to choose specific data from a collection to include in the projection. From any leaf in the resource's tree, you are able to click on a preceding node in the full path, except for the initial resource you are drawing from (for example, the **Patient** resource name) or the leaf itself (for example, the terminal **display** node). In the example below, the filter has been created by clicking on **coding**; it is usually best practice to set a filter on the deepest node of the resource tree. You can set conditions to determine which fields to show in the projection. The filter is set to include the **display** element of the item in the collection where the **code** element equals "SS."

**Patient.identifier.type.coding.display**

Set count: 413  
Data type: String  
Different values: 4

Show Histogram

Column name: IdentifierTypeCodingDisplay

Index:

Filter:  
Patient.identifier.type.coding.display

code equals SS

Add To Projection

The table projected by this transformation will show the **display** element of the **coding** objects that had a **code** of "SS." However, if the filter was set on **code** elements equal to "MR," the table will show the **display** element of the **coding** objects that had a **code** of "SS."

You can use the filter option with different operations; it is not limited to use with equals. Additional operations include **greater than**, **less than or equal to**, **greater than or equal to**, **exists**, **like regex**, **resolves to type**, and more. You should configure a filter that fits your needs.

You can also add multiple filters for a particular element to finely tune which data appears in your table; however, multiple filters cannot be set on the same tier of a path. Additionally, you cannot set a filter on elements of a subtable; instead, you must set a filter on the subtable itself, which will then include only the elements that meet the filtered condition.

Note that in some cases, a collection may not include an object that matches the filter you have specified. In these cases, the rows that do not contain such elements will not have any data in that column.

### 4.3 Adding an Object to the Schema

To add an element of an object associated with the resource, click through the object's elements until you get to a primitive element as described in "[Adding a Primitive Element](#)."

You are allowed to set a filter on objects, as you can with collections. In this case, the filter is most useful as a method to remove any values from the table that do not match a certain criteria. In the example below, the filter is used to show the

**start** element of a **CarePlan** when the **end** element is January 31, 1969. In the resultant **CarePlan** table, rows that did not end on that date will not have any data in the **start** column.

The screenshot shows a configuration window for a transformation specification. At the top, it displays the name 'CarePlan.period.start', a set count of 511, a data type of String, and more than 400 different values. A 'Show Histogram' button is located to the right. Below this, the 'Column name' is set to 'PeriodStart'. There is an unchecked 'Index' checkbox and a 'Filter' section. The filter is configured with the table name 'CarePlan.period.start', a dropdown set to 'end', an operator set to 'equals', and a date value '1969-01-31T20:42:14-05:00'. A trash icon is next to the filter. An 'Update' button is at the bottom right.

## 4.4 Viewing Transformation Specifications

On the main FHIR SQL Builder page, you will see your Transformation Specifications listed under the header. Columns in the listing provide the following information:

- **Name** – The name of this specification.
- **Analysis** – The analysis used to create this specification.
- **Description** – The description of the transformation specification as defined when it was initially created.
- **Last Modified** – The date and time the specification was last modified.
- **Actions** – Provides buttons you can use to manage the specification.
  -  **Edit** – Opens the **Edit Transformation Specification** page.
  -  **Copy** – Opens the **Copy Transformation Specification** dialog box to create a new specification identical to this one. For more information, see “[Copying a Transformation Specification](#).”
  -  **Export** – Downloads a JSON file of the transformation specification that can be shared and imported. For more information, see “[Exporting a Transformation Specification](#).”
  -  **Delete** – Deletes the transformation specification. A transformation specification cannot be deleted if a projection depends on it.

## 5 Exporting, Importing, and Copying a Transformation Specification

Transformation specifications can be exported, imported, and copied. Imported and copy transformation specifications can be further edited to further customize them.

## 5.1 Exporting a Transformation Specification

Transformation specifications can be exported to allow you to share these specifications with other users or instances, where they can be imported. To export a transformation specification, click the **Export** icon on the row of the **Transformation Specifications** table on the home page. A JSON file will be downloaded to your local file system. This file can be shared and imported into other systems.

## 5.2 Importing a Transformation Specification

A previously exported transformation specification can be re-imported on the same system or imported on another system.

To import a transformation specification, click on the **Import** button on the **Transformation Specifications** table on the home page. You will then be prompted to select the JSON transformation specification file that you want to import from your file system. When a valid file is selected, the **Import Transformation Specification** dialog will open with the following fields:

- **Name** – The name of the imported transformation specification. This will be pre-populated from the JSON file, but can be edited here.
- **Analysis** – The completed FHIR repository analysis from which to build the transformation specification. The analysis should have resources that correspond with the resources in the transformation specification.
- **Description** – A brief description to help distinguish between specifications. This will be pre-populated from the JSON file, but can be edited here.

Click the **Import** button to complete the import process. The new specification will appear in the **Transformation Specifications** table.

## 5.3 Copying a Transformation Specification

Any existing transformation specification can be copied to create a new specification with a new name and description. This new specification can be further modified and used, independent of its original specification.

To copy a transformation specification, click the **Export** icon on the row of the **Transformation Specifications** table on the home page. Fill out the dialog box as follows:

- **Name** – The name of this new specification. This will come pre-populated as “Copy of transformation,” where transformation is the name of the transformation specification that is being copied.
- **Description** – A brief description to help distinguish between specifications. Like the **Name** field, this will come pre-populated as “Copy of transformation,” where transformation is the name of the transformation specification that is being copied.

Click the **Copy** button to complete the import process. The new specification will appear in the **Transformation Specifications** table.

# 6 Create the Projection

The final step is to create a data **Projection**. Click **New** on the right side of the **Projections** section to open the **New Projection** dialog. Fill out this dialog as follows:

- **FHIR Repository** – The repository to use for this projection. This drop-down menu includes the names of the repositories added when creating a new **Analysis**.
- **Transformation Specification** – The specification used to create this projection.

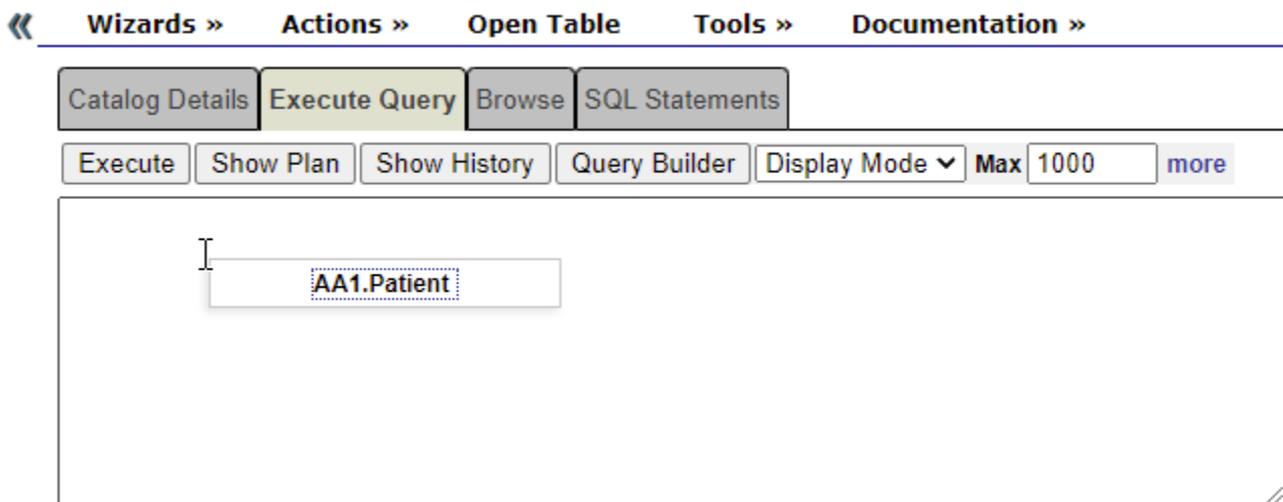
- **Package Name** – A name for the package the SQL tables will be put in.
- Click **Launch Projection** to create the projection.

Once you have created the projections, columns provide the following information:

- **FHIR Repository** – The name of the FHIR repository on which the projection is based.
- **Transformation Specification** – The Transformation Specification that the Projection is based on.
- **Namespace** – The namespace that the Projection will be stored in.
- **Package Name** – The package that the Projection will be stored in.
- **Actions** – Provides buttons you can use to manage the specification.
  -  **Link** – Opens the **System Explorer > SQL** page of the **Management Portal**.
  -  **Update** – Updates the **Projection** to reflect changes in the underlying **Transformation Specification**.
  -  **Delete** – Deletes the **Projection**.

Clicking the **Link** button opens a page in the **Management Portal**. Enter the package name you specified when creating the **Projection** into the **Filter** box on the upper left side of the screen. The projections you have created will appear under **Tables** on the left.

Drag the table to the **Execute Query** tab, as shown:



Then execute the query. The table will appear below.

ID	%ResourceID	Gender	Key
1	1	male	Patient/1
223	223	male	Patient/223
447	447	male	Patient/447
1169	1169	female	Patient/1169
1751	1751	female	Patient/1751
1969	1969	female	Patient/1969
2184	2184	male	Patient/2184
2807	2807	female	Patient/2807
3002	3002	female	Patient/3002
3996	3996	female	Patient/3996
6118	6118	female	Patient/6118
6432	6432	female	Patient/6432
8650	8650	female	Patient/8650
8941	8941	male	Patient/8941
9105	9105	female	Patient/9105
10457	10457	female	Patient/10457
11386	11386	male	Patient/11386

**Note:** A system task called **IndicesTask** will be created on the FHIR Repository if the system were to shut down while indexes were being built. This task will resume building indices when the system starts up again.

## 7 Next Steps: Tune Query Performance

Unlike a regular table, InterSystems IRIS for Health does not run the [Tune Table utility](#) automatically the first time you run a **SELECT** query on a FHIR SQL Builder data projection. The Tune Table operation calculates and stores statistics about the tabular data which the SQL compiler then uses when it selects the most efficient SQL Query Plan for a given query.

If you are not satisfied with the performance of the SQL queries you run against FHIR SQL Builder data projections, [examine the SQL Query Plan](#) for a query to determine whether InterSystems IRIS for Health is selecting the optimum approach to executing the query based on the properties of your data. To calculate and set these properties for your projections, [run the Tune Table utility](#). You can do so easily by selecting Tune Table from drop-down list of **Actions** in the Management Portal SQL interface.

For more guidance, refer to our documentation on [optimizing InterSystems SQL performance](#)