**Краткая инструкция по формированию JSON файлов на основе конструктора запросов**

**Общие идеология**

Стараемся разделить работу программиста и аналитика.

Программист на основе поставленной задачи генерирует шаблон запроса, в который добавляет все необходимые таблицы и указывает связи между ними. В общем случае для похожих задач можно сделать один шаблон со всеми, даже избыточными данными для конкретной задачи, связями

Аналитик на основании шаблонов создает экземпляры запроса, в который выбирает необходимые поля для конкретной задачи и добавляет, при необходимости, дополнительные условия.

Например:

Почти все приказы по персоналу имеют схожую структуру и для них можно сделать один шаблон, а в экземплярах запроса уже указывать тип РПД (1, 2 и т.д.)

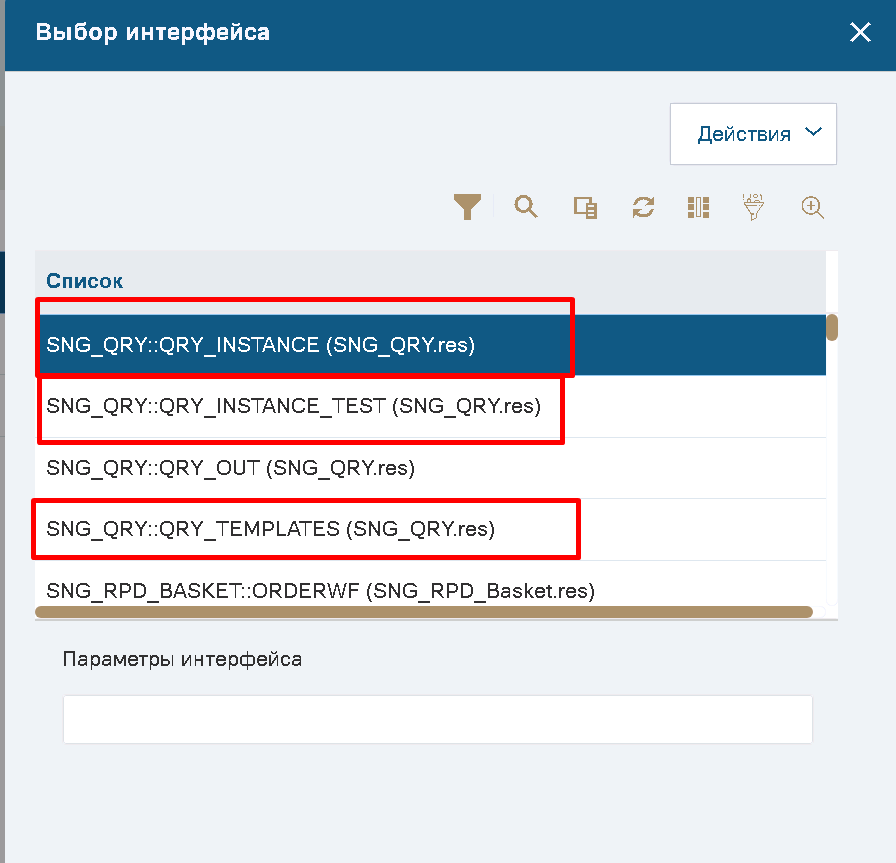
**Описание функционала.**

Интерфейсы не выведены в меню, поэтому сейчас все они запускаются через «сервис» - «утилиты» -«запуск внешнего интерфейса»

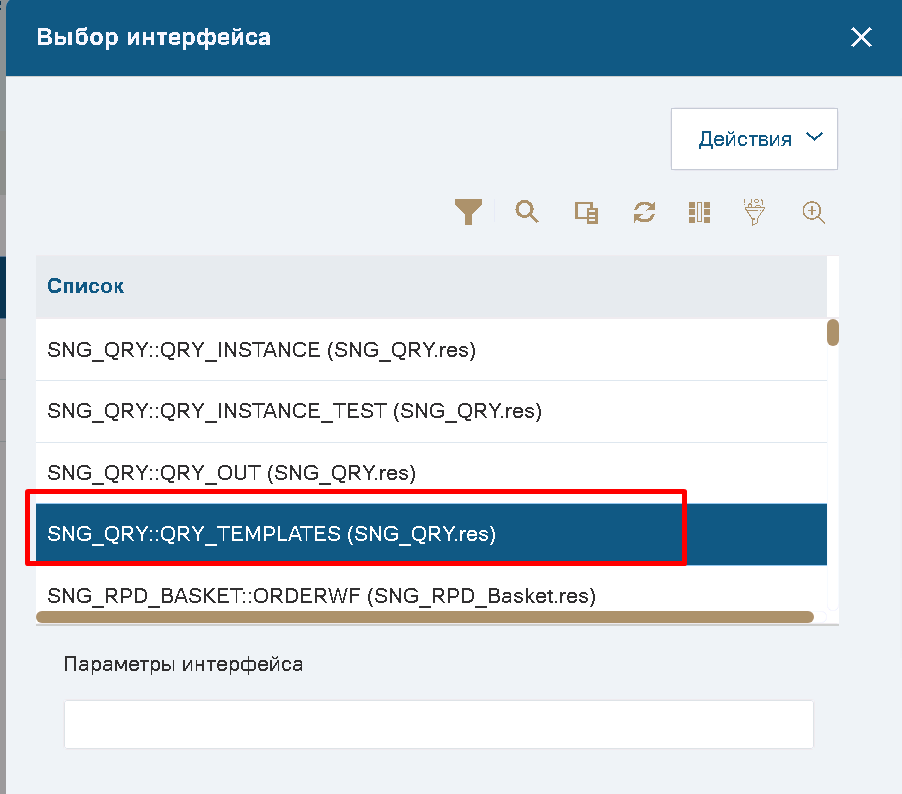
Шаблоны запросов - SNG\_QRY::QRY\_TEMPLATES

Экземпляры запросов - SNG\_QRY::QRY\_INSTANCE

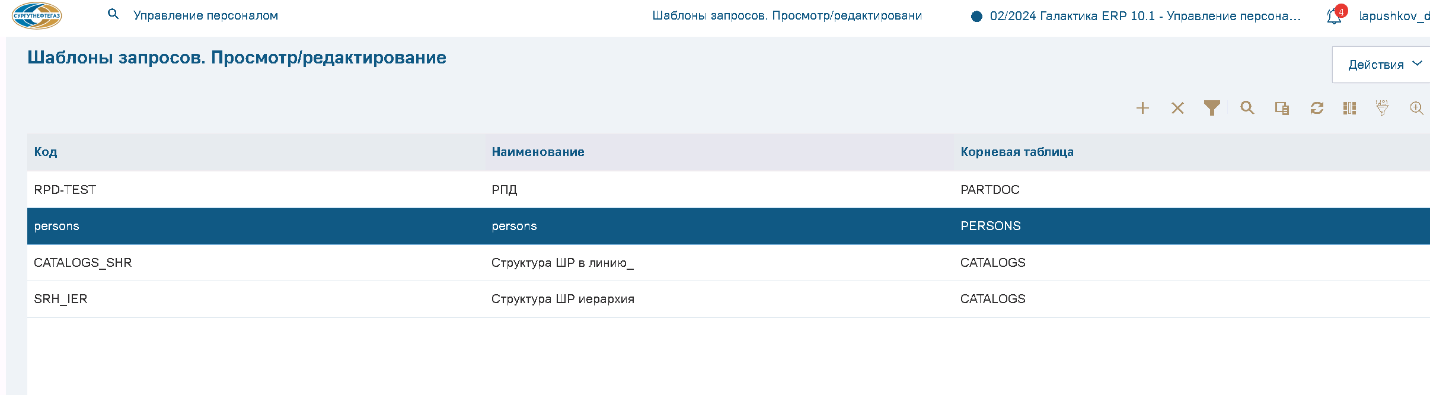
Интерфейс для тестирования - SNG\_QRY::QRY\_INSTANCE\_TEST



**Шаблоны запросов (SNG\_QRY::QRY\_TEMPLATES).**



*Список шаблонов*



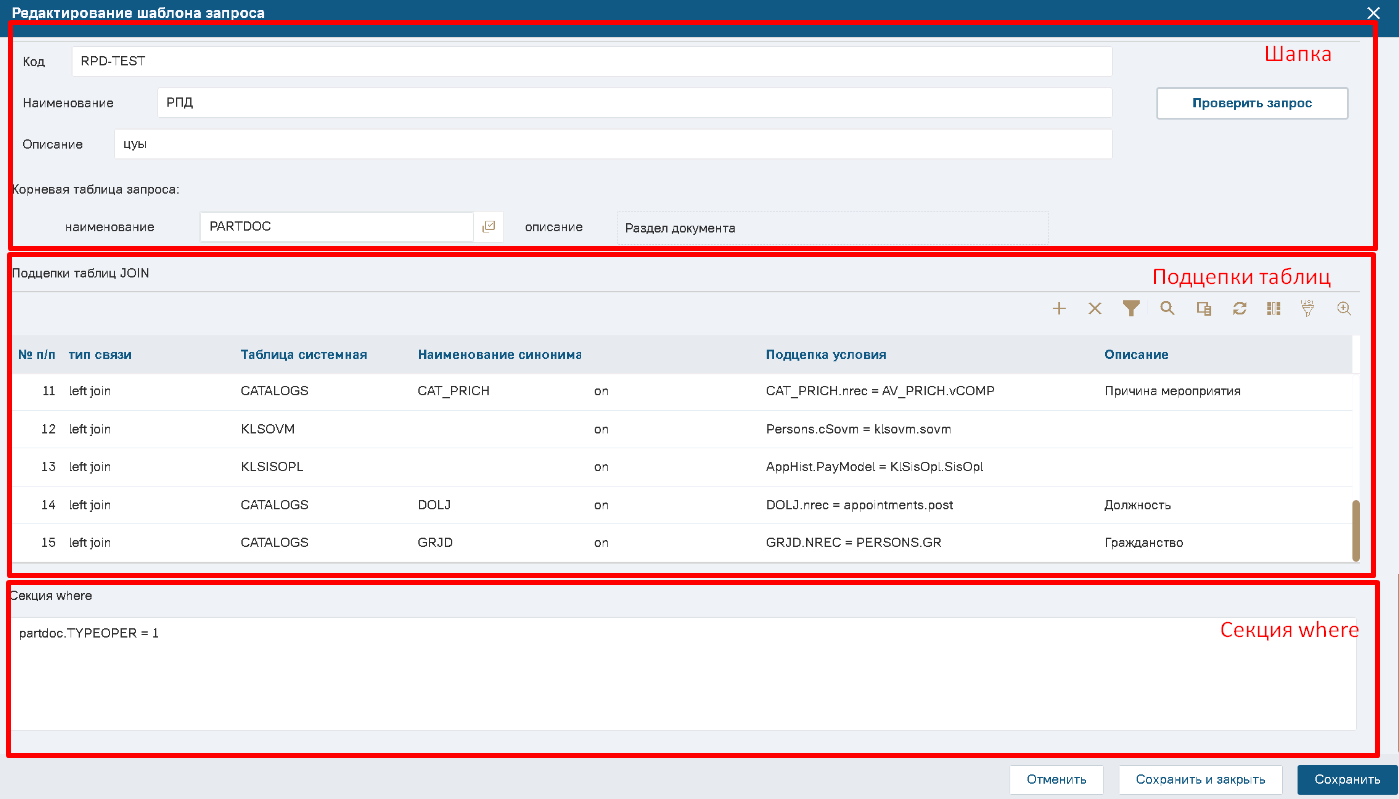
Код – строковый уникальный идентификатор шаблона (задается пользователем)

Наименование – наименование шаблона

Корневая таблица – таблица от которой строится запрос

*Карточка шаблона*

Карточка разбита на 3 части: шапка, подцепки таблиц, секция WHERE.



Шапка – указывается код, наименование, описание и выбирается корневая таблица из списка таблиц системы, также тут расположена кнопка «Проверить запрос», которая выполняет корректность созданного запроса.

Подцепки таблиц – список таблиц «сцепленных» как с корневой таблицей, так и друг с другом (по сути это секция JOIN в будущем запросе).

Описание колонок:

|  |  |
| --- | --- |
| Название колонки | Описание |
| № п/п | Номер по порядку в запросе, т.е. порядок в котором будут добавлены строки в запрос.  Заполняется автоматически., а при необходимости его можно изменить вручную.  Значение уникально в рамках шаблона |
| Тип связи | Вводится вручную JOIN, INNER JOIN, LEFT JOIN и т.д. |
| Таблица системная | Выбор из каталога таблиц системы |
| Наименование синонима | Название данной таблицы в текущем шаблоне. Этим названием необходимо пользоваться при указании условий сцепок таблиц и в секции WHERE. Также данные названия будут в экземпляре запроса, которые сделаны на основании данного шаблона |
| Подцепка условия | Условия сцепки таблиц и ограничения на таблицу |
| Описание | Произвольный текст для понимая, что это за связь, например, это будет удобно для аналитика при заполнении экземпляра запроса |

Секция WHERE

Вводится вручную. Произвольные условия/ограничения целиком на запрос. Можно использовать любые SQL операторы

На текущий момент это поле надо обязательно заполнять, если же условия не требуются или будут определены только в экземпляре, тогда можно написать условие, которое всегда верно, например, 1=1.

Проверка запроса.

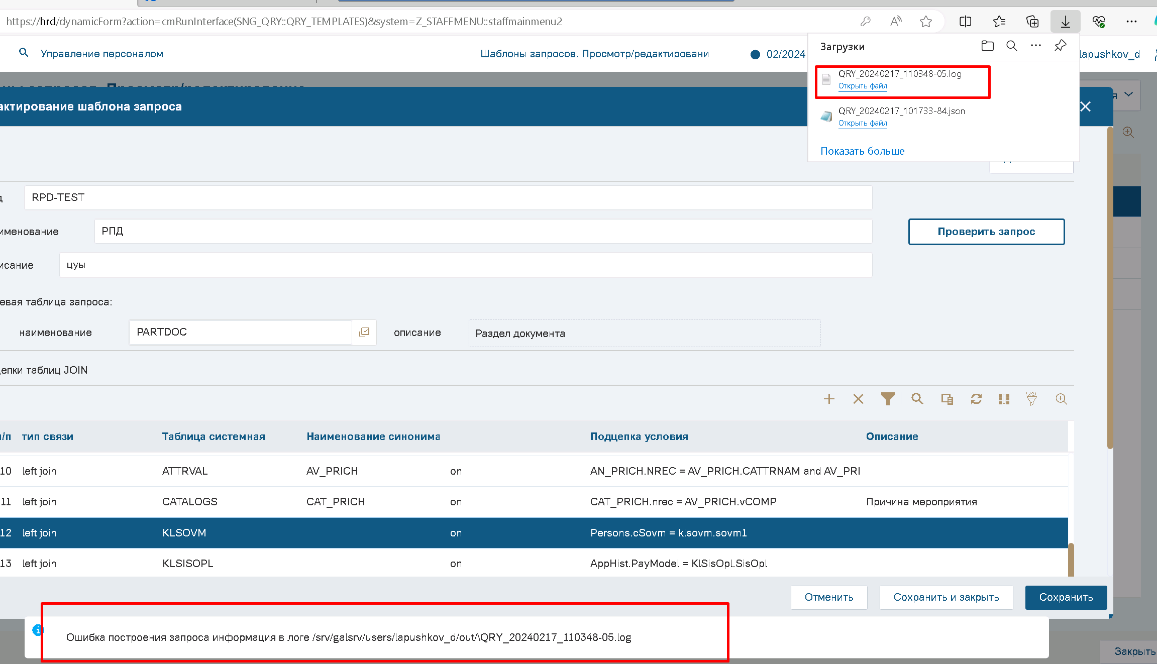
После того, как все заполнено рекомендуется проверить запрос соответствующей кнопкой в шапке шаблона.

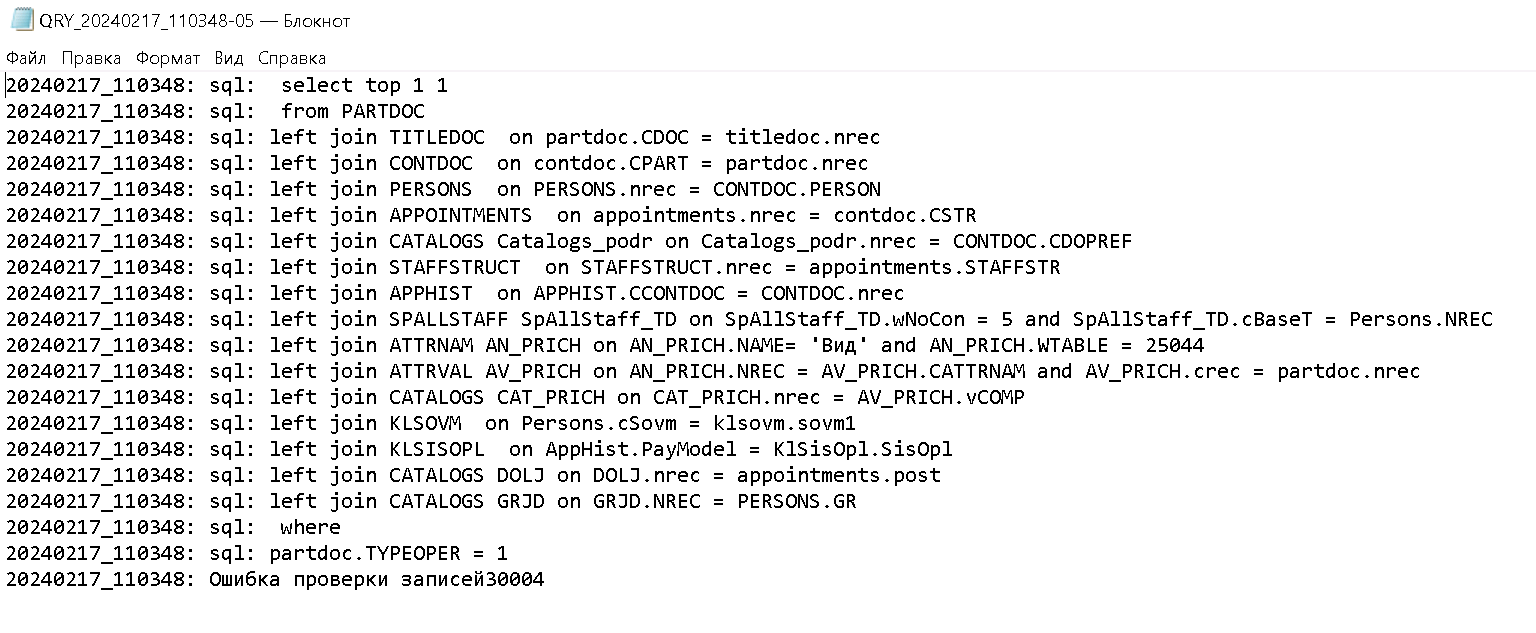
Если запрос корректный, то внизу выходит соответствующее сообщение



Если же есть ошибки, то выходит сообщение об этом и автоматически скачивается лог для анализа.

Ниже пример ошибки в запросе.

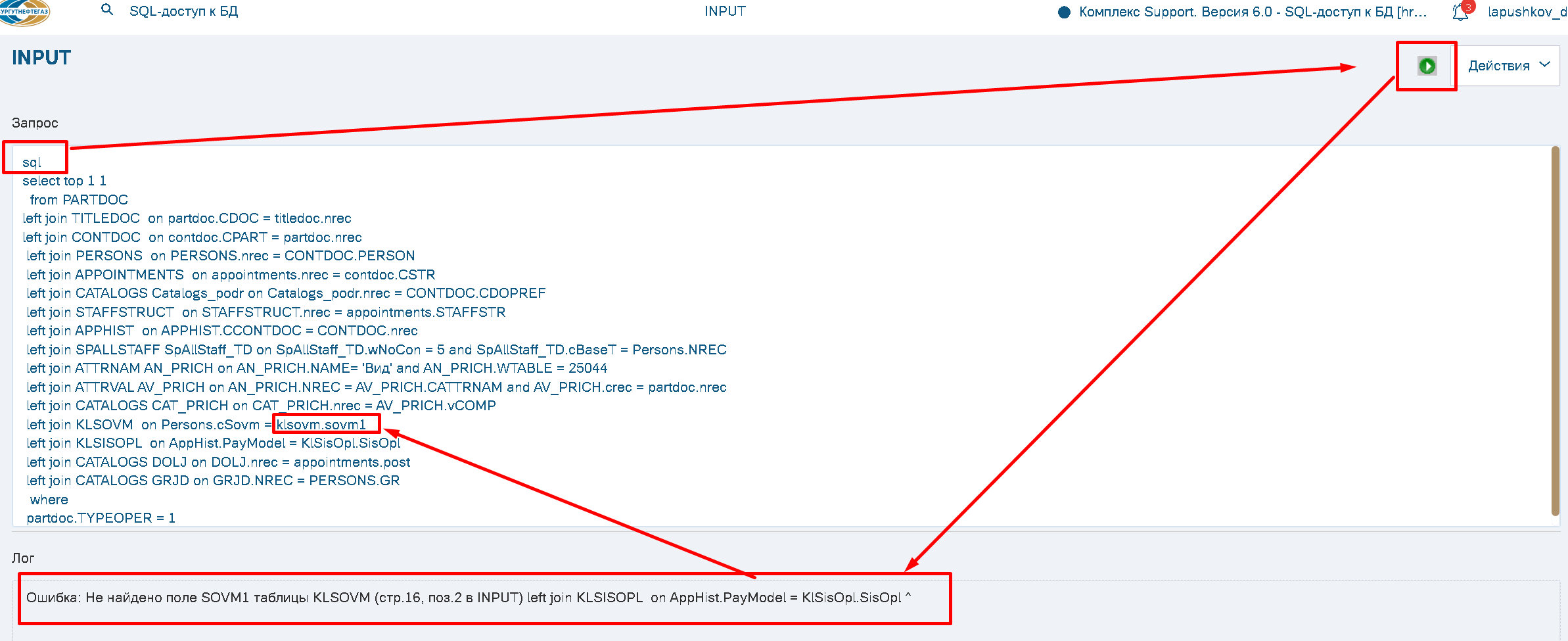




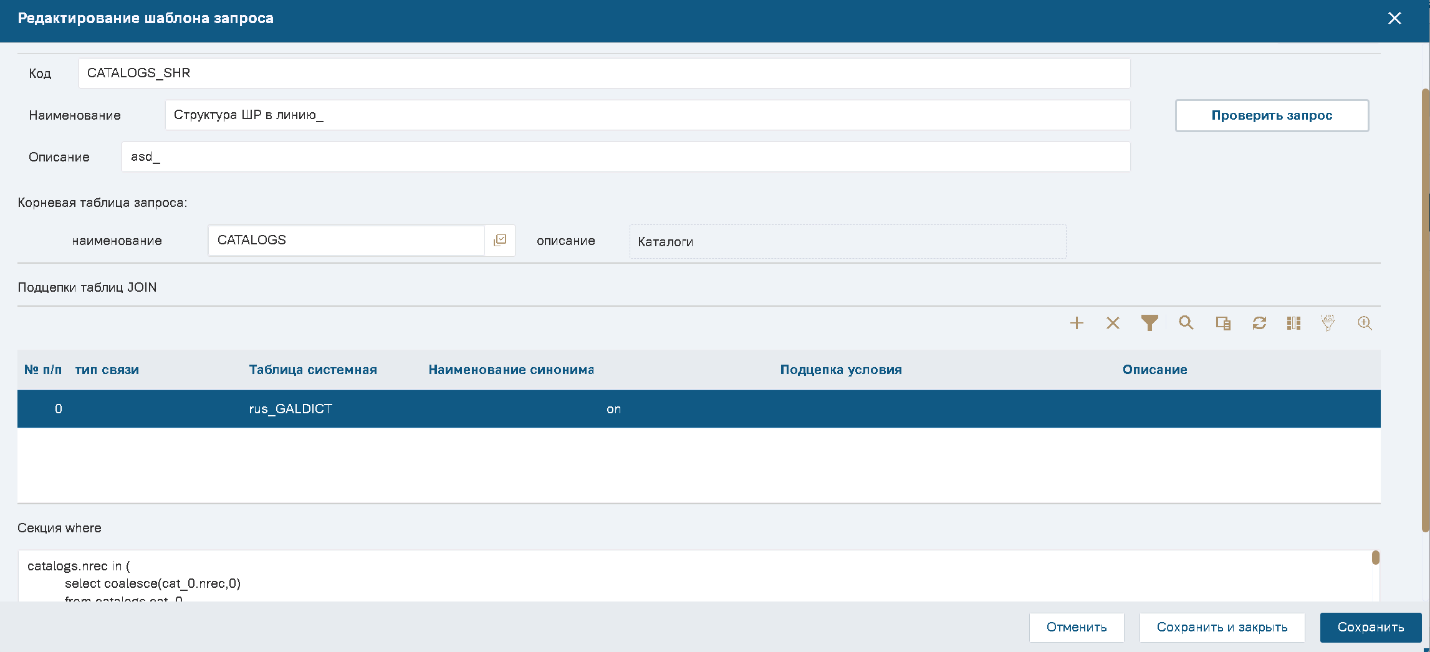
В логе содержится запрос, который был построен и отправлен на сервер СУБД.

Текст запроса можно взять и выполнить в саппорте, добавив перед запросом секцию SQL и саппорт сообщит, что не так.

В данном примере ошибка в наименовании поля таблицы KLSOVM.

В секцию WHERE можно добавлять не только простые условия, но и подзапросы

Пример шаблона на выгрузку структуру подразделений из CATALOGS, в данном примере нет подцепок таблиц (при необходимости их можно добавить).



Вот текстовка секции WHERE на 6 уровней вложенности:

catalogs.nrec in (

select coalesce(cat\_0.nrec,0)

from catalogs cat\_0

where cat\_0.syscode = -11

union all

select coalesce(cat\_1.nrec,0)

from catalogs cat\_0

left join catalogs cat\_1 on cat\_1.cParent = cat\_0.nrec

where cat\_0.syscode = -11

union all

select coalesce(cat\_2.nrec,0)

from catalogs cat\_0

left join catalogs cat\_1 on cat\_1.cParent = cat\_0.nrec

left join catalogs cat\_2 on cat\_2.cParent = cat\_1.nrec

where cat\_0.syscode = -11

union all

select coalesce(cat\_3.nrec,0)

from catalogs cat\_0

left join catalogs cat\_1 on cat\_1.cParent = cat\_0.nrec

left join catalogs cat\_2 on cat\_2.cParent = cat\_1.nrec

left join catalogs cat\_3 on cat\_3.cParent = cat\_2.nrec

where cat\_0.syscode = -11

union all

select coalesce(cat\_4.nrec,0)

from catalogs cat\_0

left join catalogs cat\_1 on cat\_1.cParent = cat\_0.nrec

left join catalogs cat\_2 on cat\_2.cParent = cat\_1.nrec

left join catalogs cat\_3 on cat\_3.cParent = cat\_2.nrec

left join catalogs cat\_4 on cat\_4.cParent = cat\_3.nrec

where cat\_0.syscode = -11

union all

select coalesce(cat\_5.nrec,0)

from catalogs cat\_0

left join catalogs cat\_1 on cat\_1.cParent = cat\_0.nrec

left join catalogs cat\_2 on cat\_2.cParent = cat\_1.nrec

left join catalogs cat\_3 on cat\_3.cParent = cat\_2.nrec

left join catalogs cat\_4 on cat\_4.cParent = cat\_3.nrec

left join catalogs cat\_5 on cat\_5.cParent = cat\_4.nrec

where cat\_0.syscode = -11

union all

select coalesce(cat\_6.nrec,0)

from catalogs cat\_0

left join catalogs cat\_1 on cat\_1.cParent = cat\_0.nrec

left join catalogs cat\_2 on cat\_2.cParent = cat\_1.nrec

left join catalogs cat\_3 on cat\_3.cParent = cat\_2.nrec

left join catalogs cat\_4 on cat\_4.cParent = cat\_3.nrec

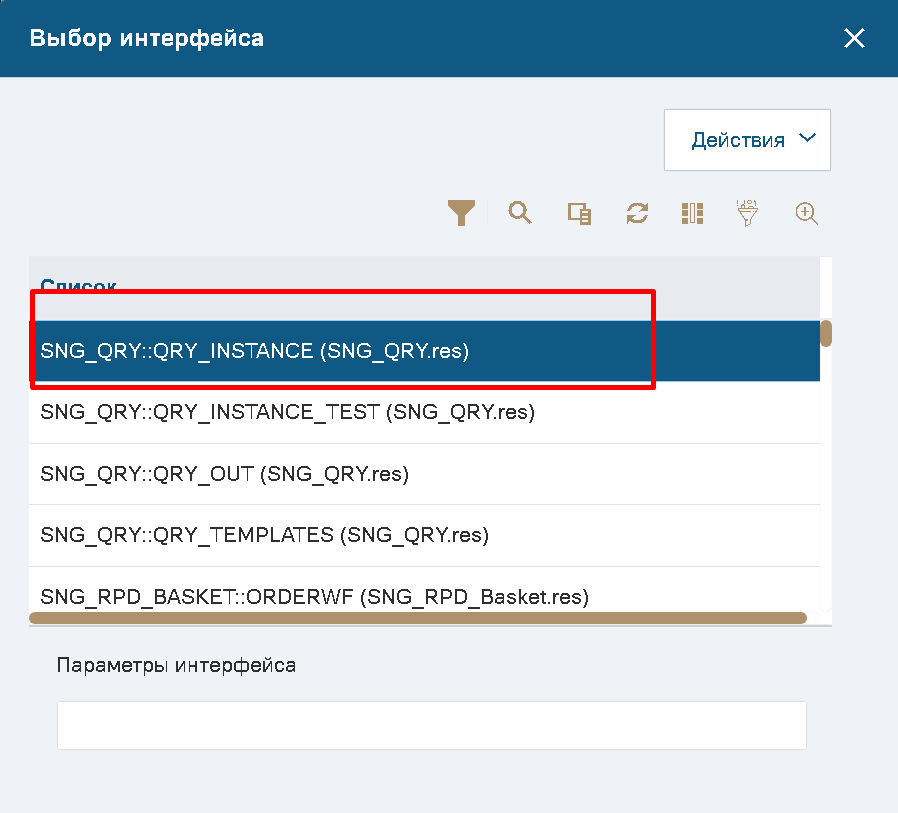
left join catalogs cat\_5 on cat\_5.cParent = cat\_4.nrec

left join catalogs cat\_6 on cat\_6.cParent = cat\_5.nrec

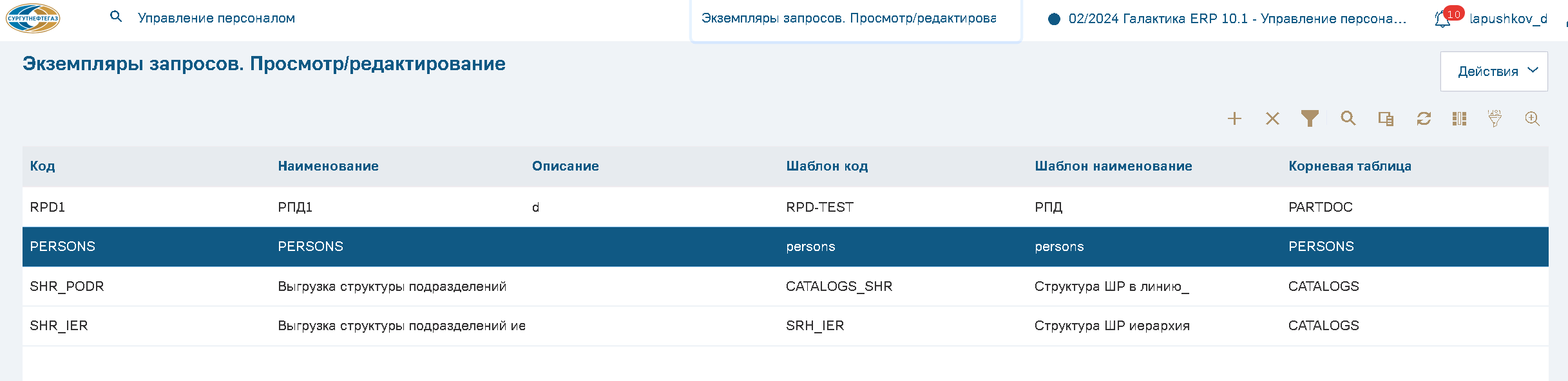
where cat\_0.syscode = -11

)

**Экземпляры запросов - SNG\_QRY::QRY\_INSTANCE**



*Список экземпляров запросов.*



Код – строковый уникальный идентификатор запроса (по этому коду вызываются функции формирования JSON)

Наименование – наименование экземпляра

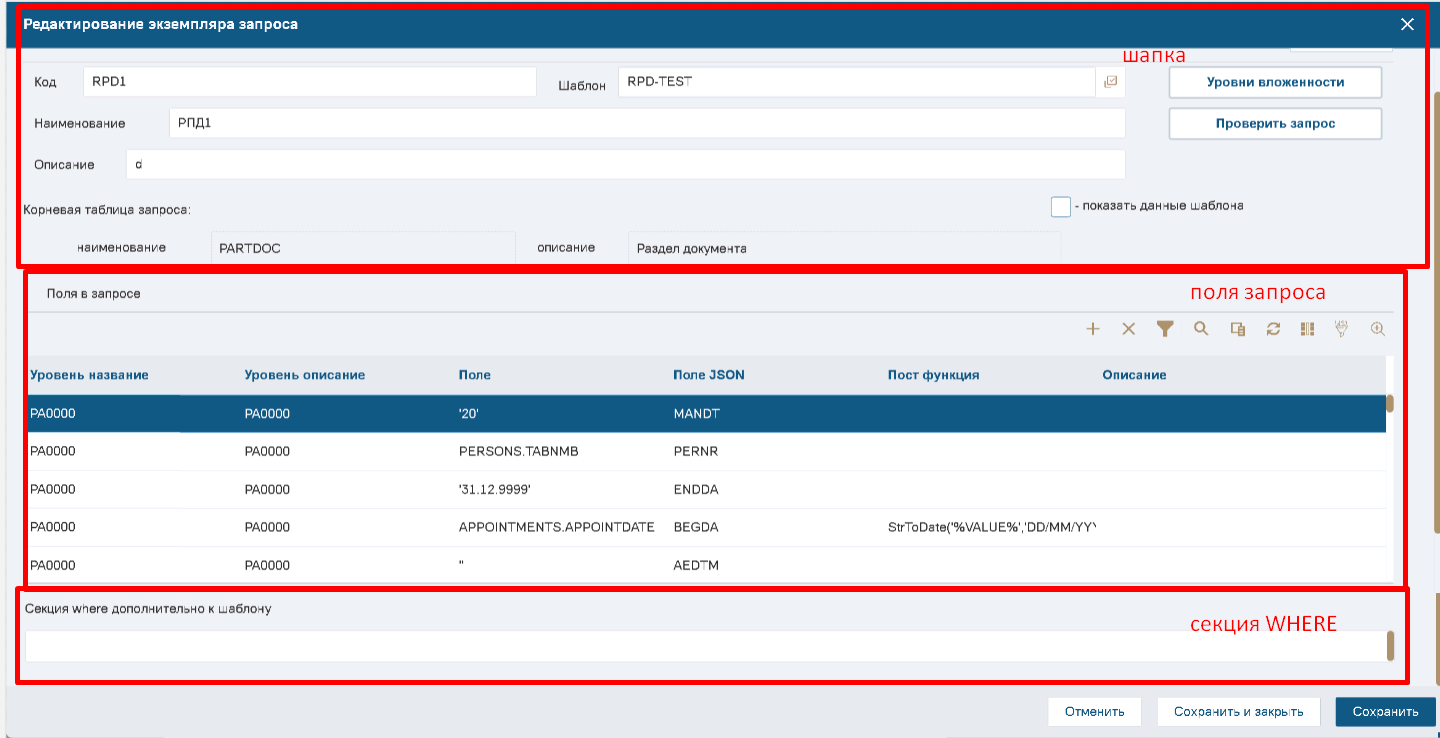
Описание – описание экземпляра

Шаблон код и шаблон наименование – данные из шаблона

Корневая таблица – из шаблона

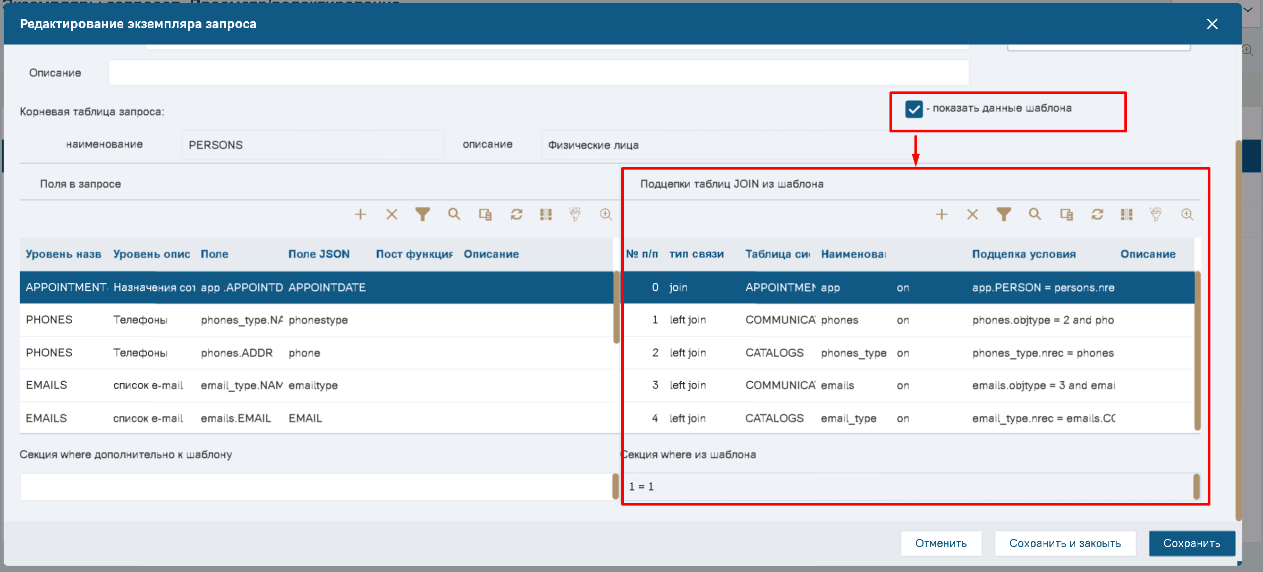
Карточка экземпляра запроса

Карточка экземпляра похожа на карточку шаблона и также состоит из трех частей: шапка, список полей, секция WHERE.



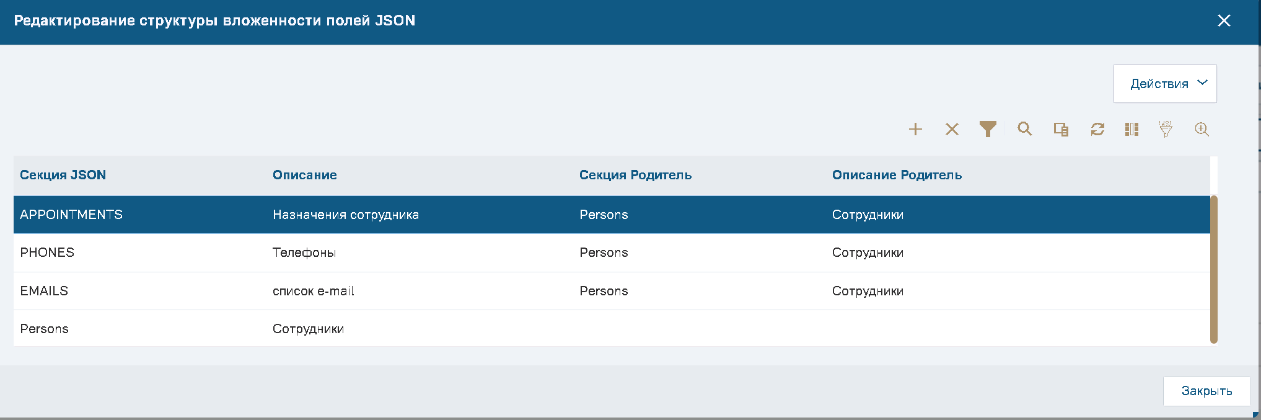
В шапке указывается код, наименование, описание экземпляра, выбирается шаблон запроса.

При нажатии на галочку «показать данные шаблона» в средней панели появляется дополнительная секция, где отображена структура шаблона.

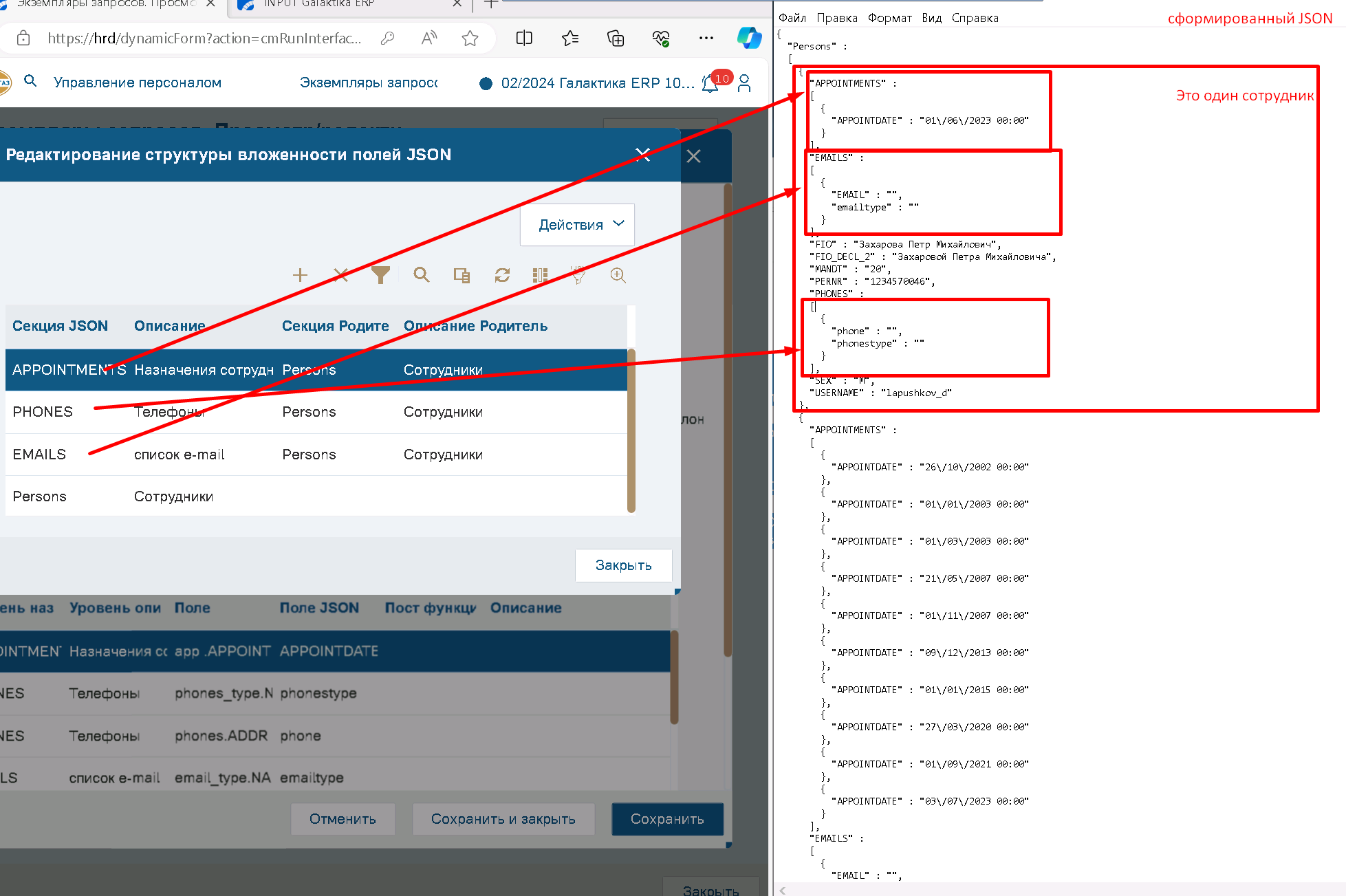


Также в шапке есть две кнопки: «Уровни вложенности» и «Проверить запрос».

При нажатии на «Уровни вложенности» открывается окно редактирования уровней



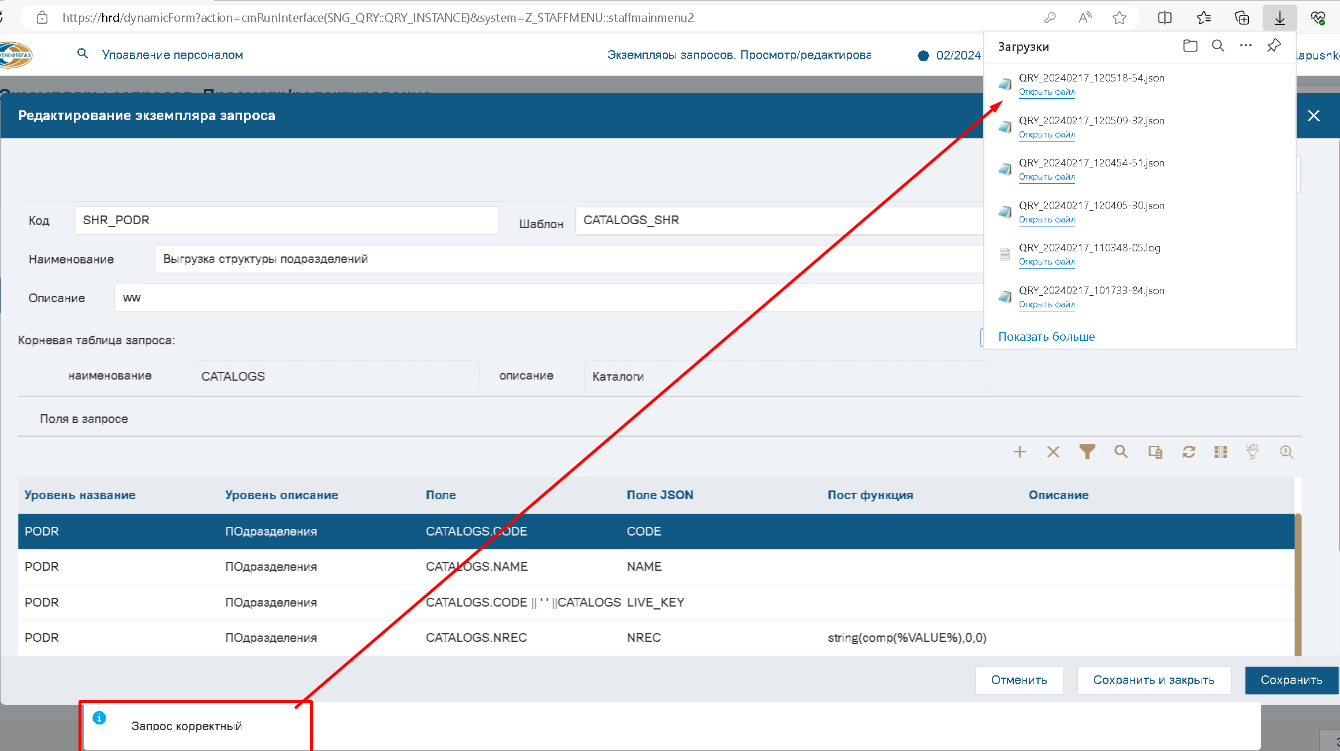
Уровни вложенности: это по сути названия секций в исходящем JSON, которые могут вложены друг в друга. На скрине показаны уровни вложенности для запроса PERSONS, скрин ниже показывает как будет выглядеть JSON, т.е. для каждого сотрудника в JSON будут сформированы вложенные секции APPOINTMENTS, PHONES, EMAILS.



Уровни вложений не ограничены. Однако проверок на «закольцовывание» нет, поэтому будьте внимательны

При нажатии на «Проверить запрос» выполняется полностью весь запрос и формируется JSON файл, по сути это полноценная работа функционала, тут вы сразу увидите итоговый результат работы.

Если проверка прошла успешна, то выводится соответствующее сообщение и автоматически скачивается итоговый JSON.



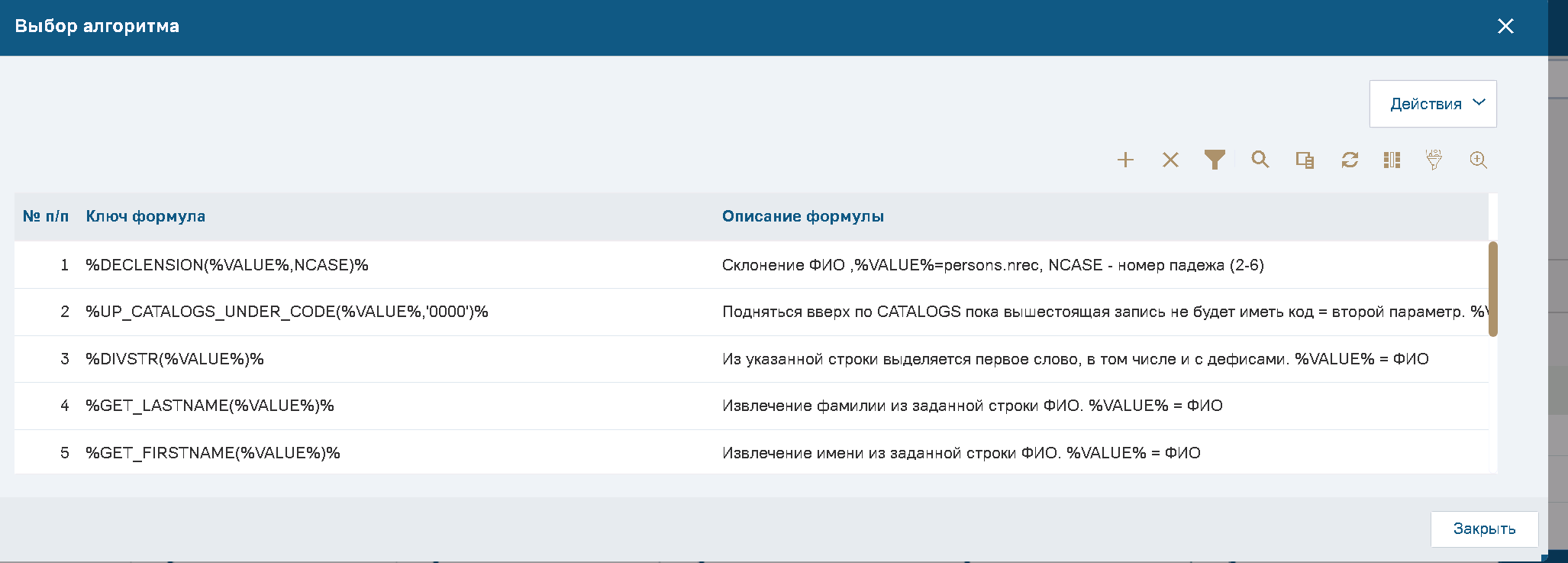
Если проверка неуспешна, но выйдет соответствующее сообщение и автоматически скачается лог для разбора ошибок.

*Поля в запросе.*

Описание колонок

|  |  |
| --- | --- |
| Название колонки | Описание |
| Уровень название, Уровень описание | Выбираются из каталога уровней вложенности, внесенных под кнопкой «Уровни вложенности». У каждой строки должен быть выбра уровень |
| Поле | Выбирается из списка полей таблиц, выбранных в шаблоне или вносится клавиатуры. |
| Поле JSON | Название поля в структуре JSON. Поле должно быть уникально в разрезе одного уровня вложенности |
| Пост функция | Зачастую запросом не все значения можно получить. Например, склонение ФИО, для этого используются пост-функции, которые дополнительно обрабатывают значения.  Поле можно вносить вручную, доступен выбор из преднастроенного каталога функций. В этом поле работают любые стандартные функции системы  Для передачи значения поля из запроса используется %VALUE%  Примеры:  StrToDate('%VALUE%','DD/MM/YYYY') – преобразование строки в дату  if(%VALUE%=1,94,12) если значение в поле равно1, то замени его на 94, иначе на 12  %UP\_CATALOGS\_UNDER\_CODE(%VALUE%,0000)% - преднастроенная функция, которая поднимается до подразделения с кодом «0000» и возвращает значение кода подразделения под ним, в качестве входного параметра идет код текущего подразделения |
| Описание | Текстовое описание для пользователя |

Список преднастроенных функций (список расширяемый)



Секция WHERE

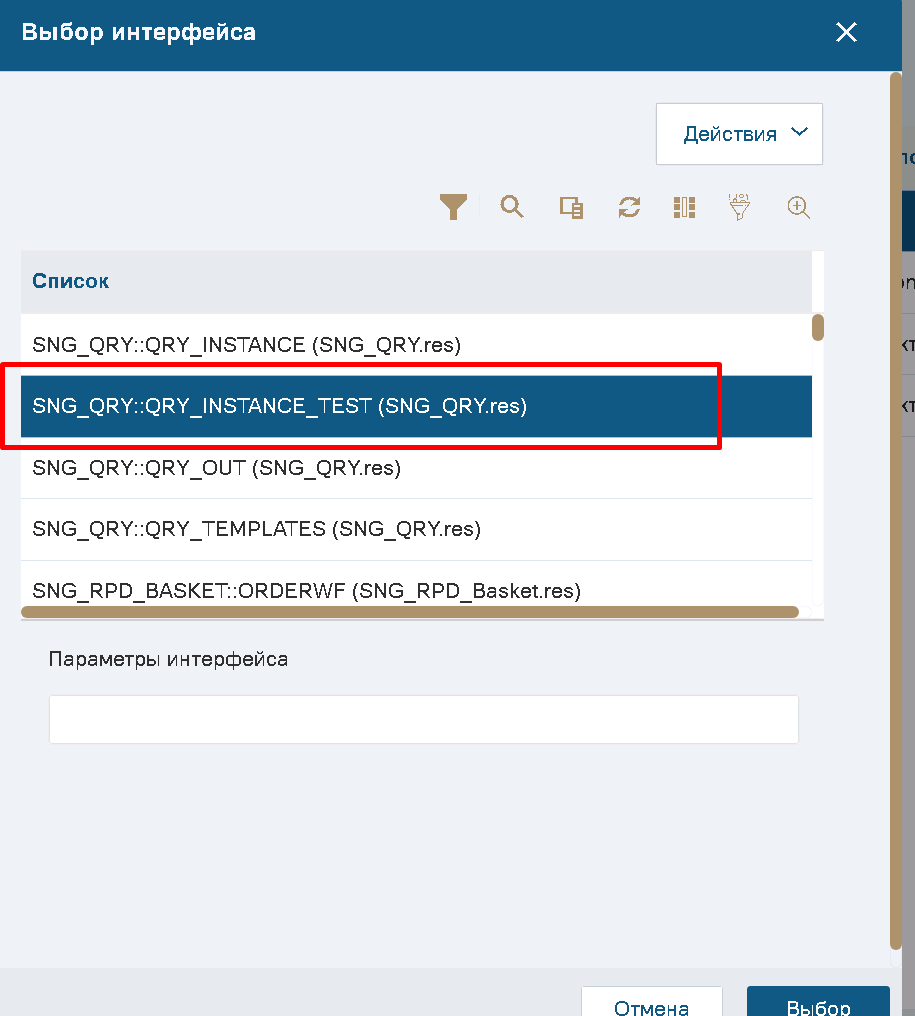
В данное поле вносятся дополнительные условия

Следует учитывать, что это не отдельное условие, а продолжение предыдущего из шаблона

Т.е. написание в секции «catalogs.code = ‘123’» является ошибкой, надо писать “and catalogs.code = ‘123’”

Разумеется вместо «and» вы будете писать нужное вам условие.

**Интерфейс для тестирования - SNG\_QRY::QRY\_INSTANCE\_TEST**

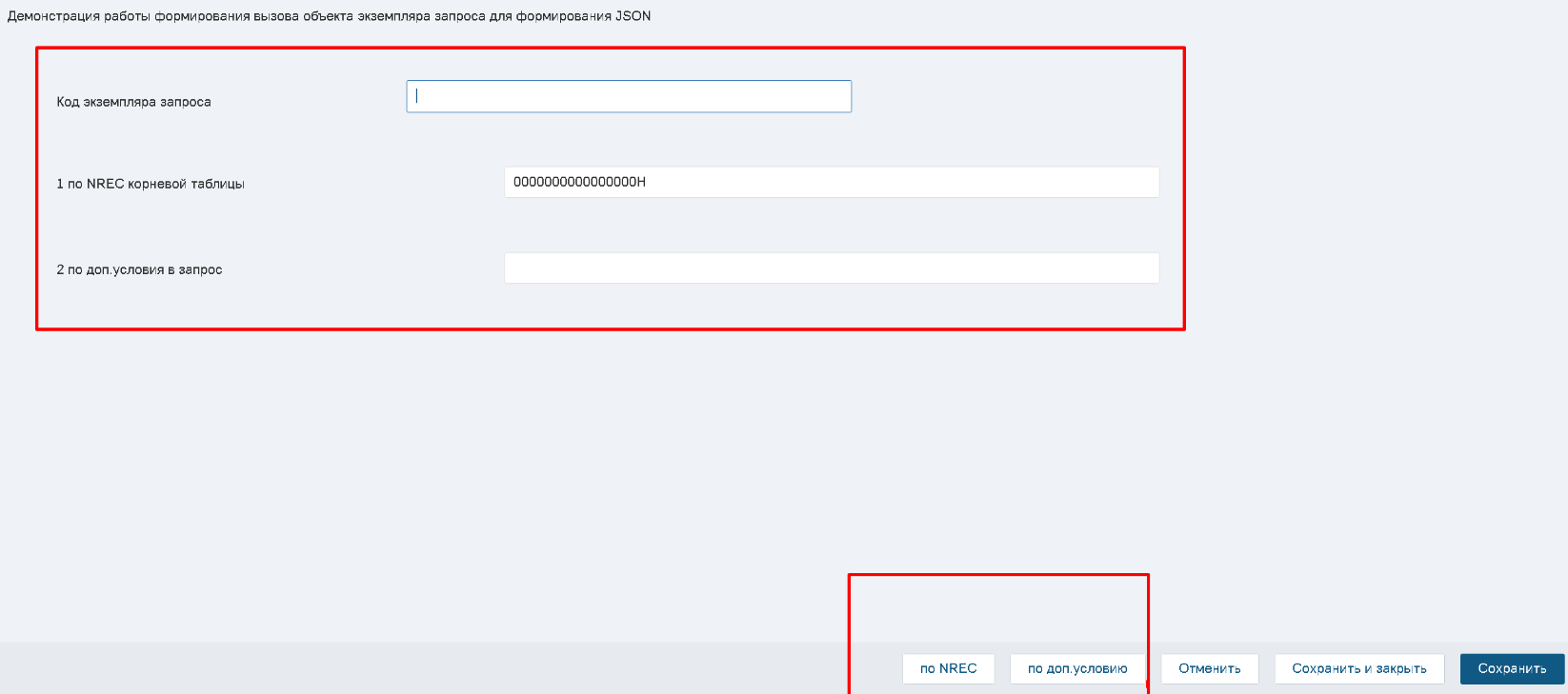


Интерфейс содержит всего 3 поля, которые вводятся вручную

* Код экземпляра запроса
* NREC корневой таблицы
* По доп.условия в запрос

И 2 кнопки

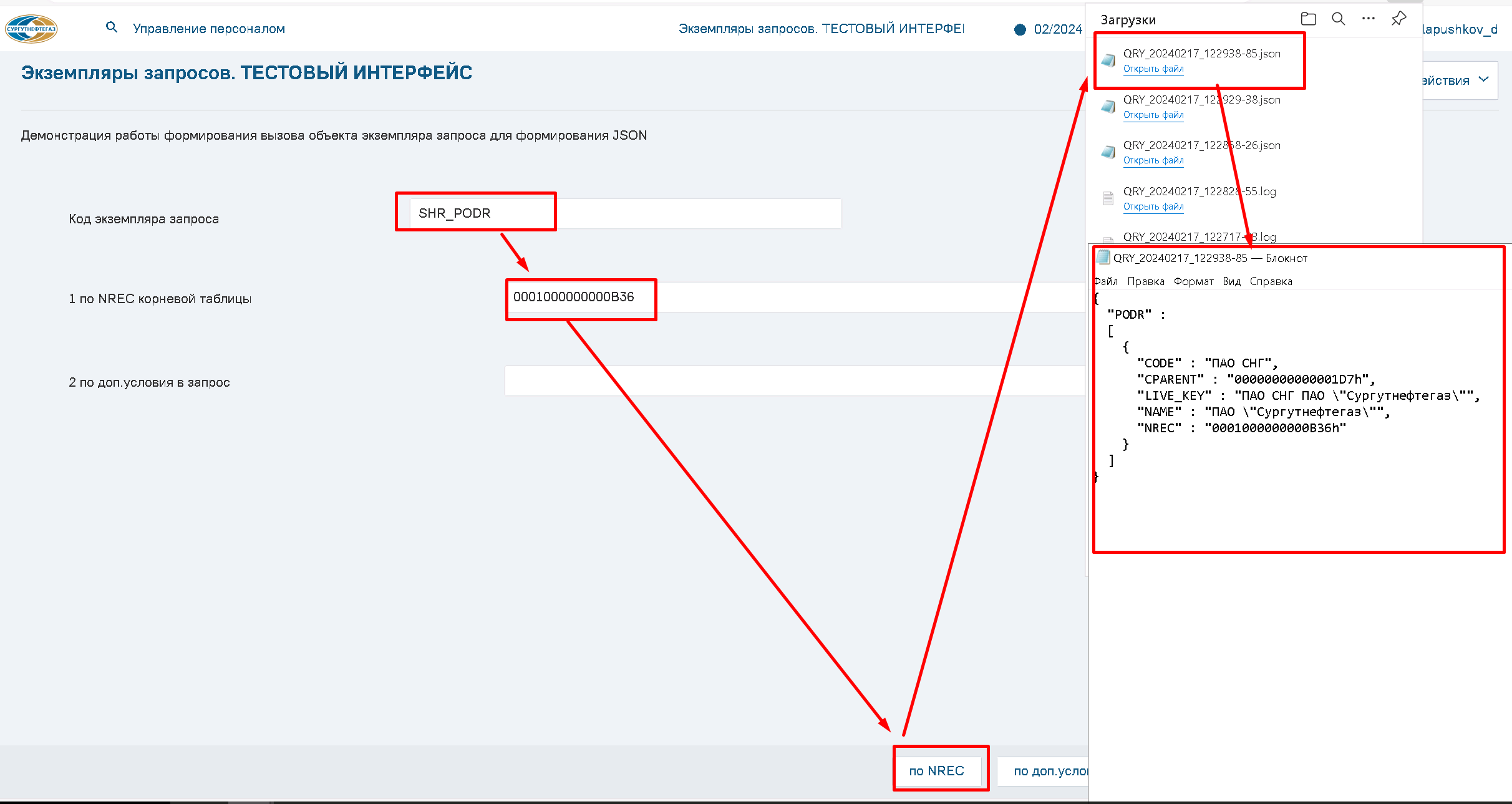
«По NREC» и «по доп.условию»



При запросе про NREC условие выборки заключается в скобки и после них добавляется «корневая таблица.NREC»= вставленный NREC

При запросе по «доп. Условия в запрос» условие выборки заключается в скобки и после них добавляется указанное условие

Пример запроса по NREC корневой таблицы



Пример запроса по доп условию

