
Руководство по установке и эксплуатации

Выпуск 9.0.0

pgCodeKeeper

дек. 07, 2023

1	Вступление	3
2	Установка	5
2.1	Eclipse Marketplace	5
2.2	Сайт обновления	6
2.3	Самостоятельная сборка pgCodeKeeper	8
3	Создание нового проекта	9
4	Импорт проекта	11
5	Структура проекта	13
6	Конвертация проекта	15
7	Редакторы и панели	17
7.1	Редактор проекта pgCodeKeeper	17
7.2	Поиск объекта в таблице различий	19
7.3	Фильтры объектов	19
7.4	SQL редактор	21
7.5	Просмотр зависимостей объекта БД	21
7.6	Переопределенные объекты	24
7.7	Результат запроса	24
8	Глобальные настройки	25
8.1	Основные настройки pgCodeKeeper	25
8.2	Исключенные объекты	26
8.3	Обновление БД	28
8.4	Отчеты об использовании	30
8.5	Редактор проекта	31
8.6	Редактор SQL	33
8.7	Форматирование	34
8.8	Подсветка синтаксиса	35
8.9	Шаблоны редактора SQL	35
8.10	Хранилище БД	36
8.11	Тип соединения	42
9	Настройки проекта	45

9.1	Основные настройки проекта	45
9.2	Списки исключений	46
9.3	Исключенные схемы	48
9.4	Библиотечные зависимости	49
9.5	Обновление БД	52
10	Обновление проекта	53
11	Обновление удаленной базы данных	55
12	Выбор источника базы данных	59
12.1	Группировка источников базы данных	61
13	Сортировка списка изменений	63
14	Редактирование pgCodeKeeper проекта	65
15	Форматирование	69
16	Рефакторинг	71
16.1	Быстрые исправления	72
17	Поиск объекта	75
17.1	Поиск ссылок на объекты	75
18	Сравнение схем БД	77
19	Генерация тестовых данных	79
20	Список исключаемых объектов	83
20.1	Синтаксис списков	84
20.2	Совместное использование черного и белого списков	90
20.3	Примеры работы с файлом <code>.pgcodekeeperignore</code>	91
21	Исключение схем при загрузке	95
21.1	Выбор объектов	96
21.2	Пример использования исключения схем при загрузке	96
22	CLI версия	99
22.1	Начало работы	99
22.2	Примеры	100
23	Windows аутентификация	101
23.1	Конфигурация системы	102
24	Вклад участников	103
24.1	Идея	103
24.2	Управление	103
24.3	Разработка	103
24.4	Дизайн и вёрстка сайта	104
24.5	Перевод документации	104
25	О pgCodeKeeper	105
25.1	Карточка организации	105
25.2	Свидетельство о регистрации	107

Плагин pgCodeKeeper предназначен для сравнения схем БД PostgreSQL и выборочного применения различий, с учетом возможных зависимостей между объектами схемы.

Работа с pgCodeKeeper осуществляется с помощью проекта. Он представляет собой объекты схемы БД, хранимые в отдельных файлах в иерархической структуре.

С помощью проекта возможно просмотреть структуру объектов-файлов, и создать скрипт наката схемы проекта на БД, или наоборот, модифицировать структуру проекта объектами из какой-либо БД.

Исходный код pgCodeKeeper находится в свободном доступе и доступен на [GitHub](#).

Вступление

В 2013 году, когда я приступил к работе в ООО «Технология», компания уже разрабатывала программное обеспечение для одного из крупнейших в России сервисов заказа такси «Максим». И одной из целей на тот период был переход в использовании от сервера баз данных MSSQL на PostgreSQL.

Мы начали процесс миграции с одной системы управления базами данных на другую и обнаружили две существенных особенностей PostgreSQL.

Первое, набор инструментов, как для разработки БД, так и для её сопровождения, на тот момент значительно уступал относительно хорошо известного к тому времени MSSQL.

Второе, архитектура существующей информационной системы включала в себя значительный пласт бизнес-логики приложений расположенной в базе данных. Это и представления, связанные со множеством таблиц, и хранимые процедуры.

Мы столкнулись с существенной сложностью при миграции, т.к. в отличие от СУБД подобных MSSQL и Oracle, которые при пересоздании объекта с зависимостями выполняли инвалидацию зависимых объектов, с возможностью выполнения дальнейшей перекомпиляции невалидных объектов, PostgreSQL поступал совсем по-другому.

Нашим настоящим кошмаром стал ад зависимостей между объектами БД. При пересоздании объекта, в случае, если на него ссылались другие объекты БД приходилось удалять все зависимые объекты и создавать их повторно после пересоздания «родительского» объекта.

Тогда и родилась идея о создании собственного продукта способного решить в первую очередь проблему при формировании скрипта миграции с зависимыми объектами, а более того способного облегчить работу разработчика и другими способами, такими как: автодополнение кода, навигация по коду, формирование скриптов с тестовыми данными для существующих таблиц и многое другое.

На текущий момент pgCodeKeeper решает многие из поставленных целей. Но мы не останавливаемся. На очереди – как улучшение существующих возможностей, так и достижение новых целей.

Мы работаем над тем, чтобы pgCodeKeeper мог выступать в качестве инструмента для выполнения рефакторинга объектов базы данных, улучшение работы с кодом pl/pgSql, и извлечением и отображением данных.

Теперь, когда у нас появился pgCodeKeeper, работа с PostgreSQL стала значительно проще. В конце 2017 года было принято решение вывести код продукта под одну из существующих лицензий с откры-

тым исходным кодом. Мы остановились на лицензии Apache 2.0. И именно благодаря этому, теперь pgCodeKeeper стал доступен множеству участников по всему миру.

Я благодарю вас за интерес, который вы проявили к нашему продукту и желаю вам приятной работы с pgCodeKeeper.

Руководитель отдела разработки ПО, г. Набережные Челны
Саушкин Андрей, 2018

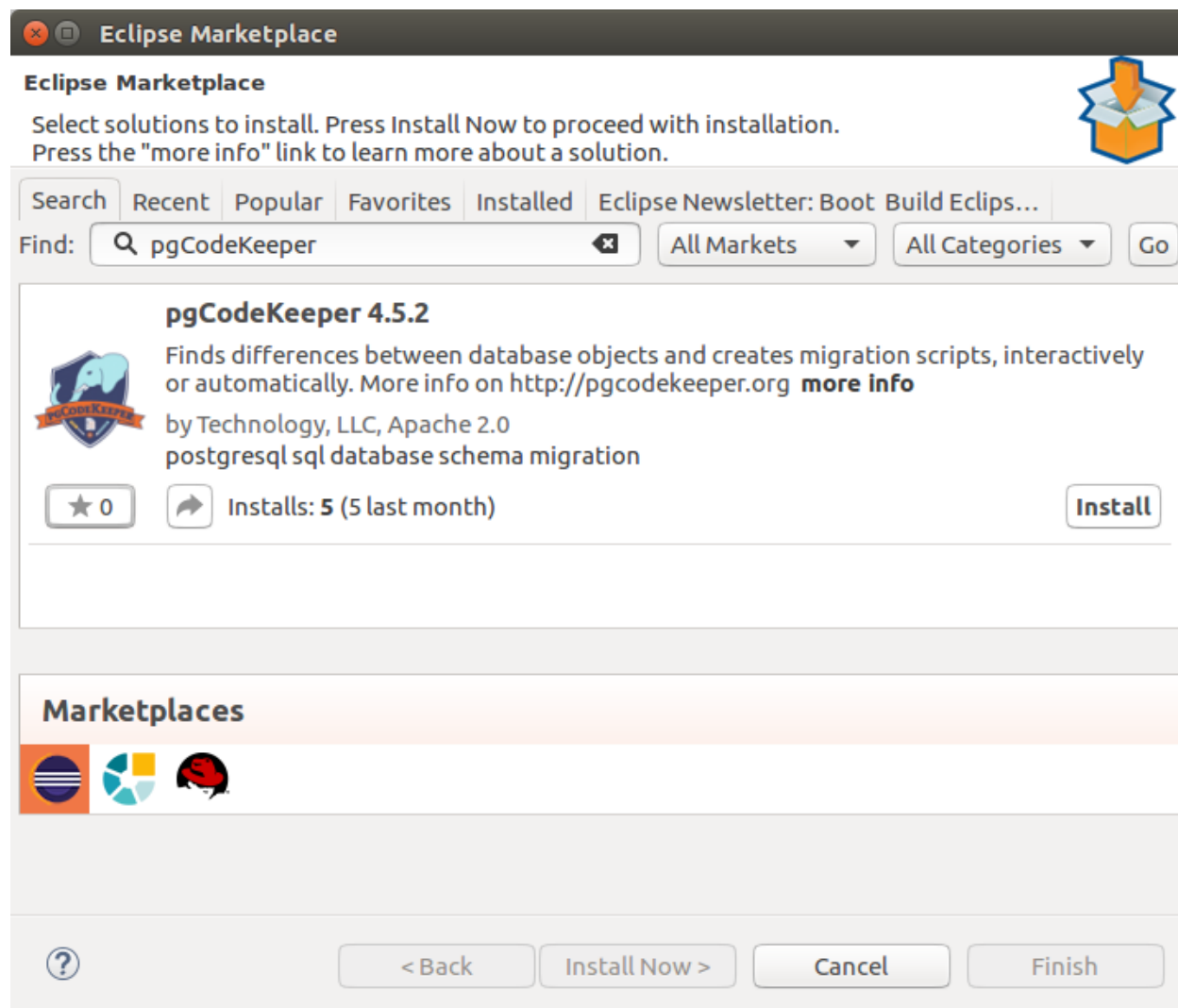
Для начала необходимо установить [Java SDK](#) версии 11+ для вашей платформы. Последние версии Eclipse требуют для работы версию 17+

2.1 Eclipse Marketplace

Самый простой путь установить pgCodeKeeper - Eclipse Marketplace : в запущенном экземпляре Eclipse версии 4.6 или выше с установленным Eclipse Marketplace, выбрать **Help > Eclipse Marketplace...**

В поле поиска ввести: pgCodeKeeper.

Выберите пакет pgCodeKeeper, ознакомьтесь с пользовательским соглашением, нажмите кнопку Finish, и начнется установка последней версии pgCodeKeeper.



По завершению установки для применения изменений необходимо перезапустить Eclipse.

Запустите Eclipse и выберите **Window > Perspective > Open Perspective > Other > pgCodeKeeper**.

Для установки предыдущих версий pgCodeKeeper перейдите по ссылке <https://github.com/pgcodekeeper/pgcodekeeper/releases> и скачайте нужную вам версию релиза и установите его вручную. На сайте представлены архивы релизов версии 5.10.5 и выше.

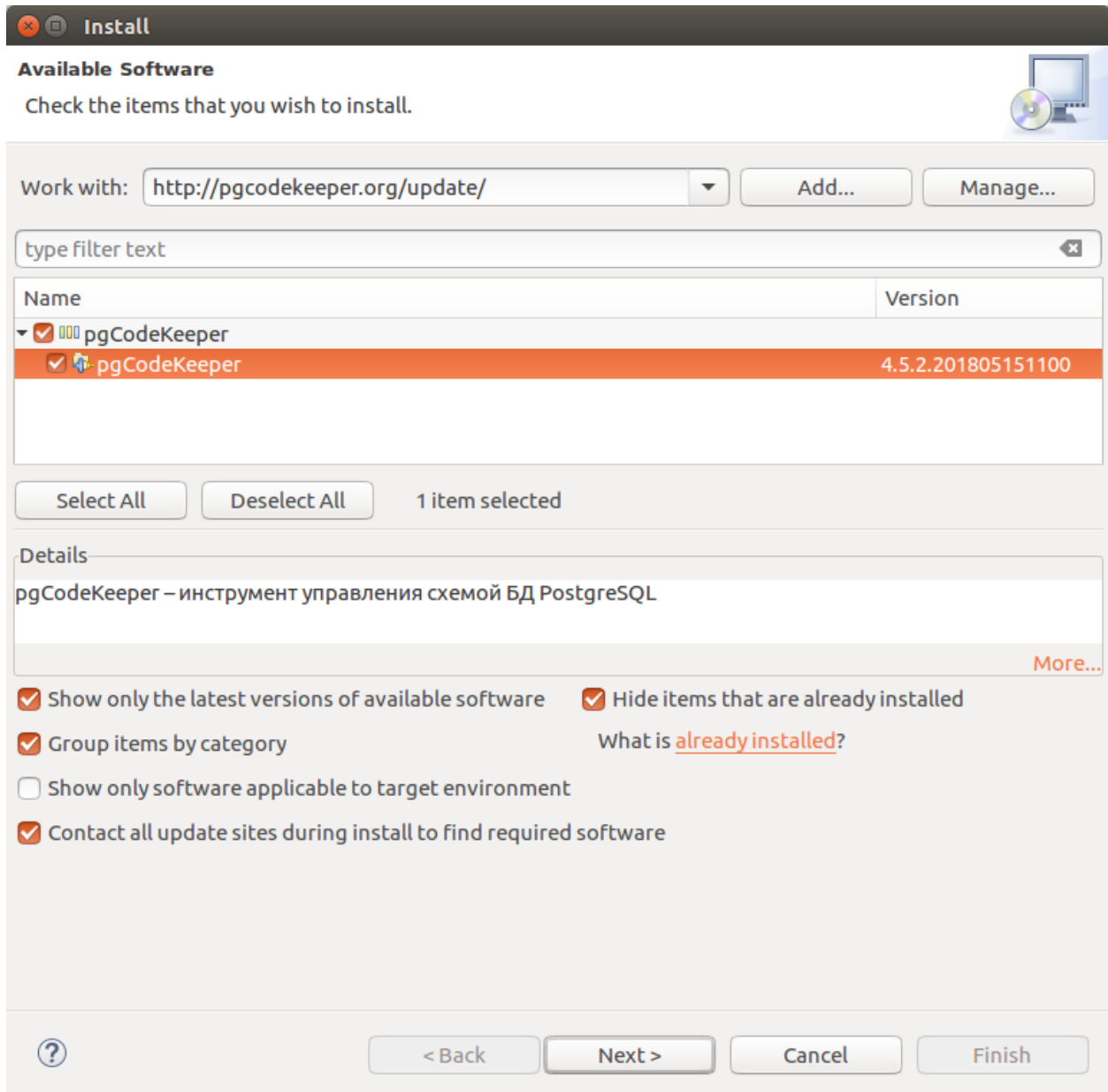
2.2 Сайт обновления

Альтернативный путь установки pgCodeKeeper – сайт обновлений: в запущенном экземпляре Eclipse версии 4.6 или выше, выбрать **Help > Install New Software...** Откроется мастер установки плагинов средствами сайта обновлений.

В поле **Work With** ввести путь к сайту обновлений: <http://pgcodekeeper.org/update/>.

После нажатия **Enter** в списке ниже появятся доступные для установки плагины.

Выберите пакет pgCodeKeeper, согласитесь с установкой, ознакомьтесь с пользовательским соглашением, нажмите кнопку **Finish** и начнется установка.



По завершению установки для применения изменений необходимо перезапустить Eclipse.

Запустите Eclipse и выберите **Window > Perspective > Open Perspective > Other > pgCodeKeeper**.

2.3 Самостоятельная сборка pgCodeKeeper

Ещё одной альтернативой является самостоятельная сборка pgCodeKeeper, которую можно скачать [здесь](#).

Создание нового проекта

- Открываем мастер создания нового проекта: **File -> New -> Project...**
- Выбираем **pgCodeKeeper Project** для работы с PostgreSQL или **pgCodeKeeper MS SQL Project** для работы с MS SQL в категории **pgCodeKeeper**. Нажимаем кнопку **Next**.
- Указываем имя проекта.
- При необходимости изменяем местоположение и рабочую область проекта.
- Нажимаем кнопку **Next**.

New pgCodeKeeper project

Project initialization
Choose a DB schema source for project initialization

Initialize project

☒ Initialize project using database ☐ Bind project to database connection

DB Schema Source: postgres_10

Project charset: UTF-8

DB connection timezone: UTC

? < Back Next > Cancel Finish

- Выбираем источник базы данных или отключаем параметр **Initialize project using database** / **Инициализировать проект базой данных**.
- Выбираем параметр **Bind project to database connection** / **Привязать проект к подключению к БД**, если хотим работать только с одной базой данных.
- Выбираем кодировку проекта.
- Выбираем временную зону подключения к БД (имеется возможность получить из текущей БД). Данный шаг не доступен для проектов MS SQL.
- Нажимаем кнопку **Finish**.

После короткого ожидания получаем новый проект и будет автоматически открыт *Редактор проекта pgCodeKeeper*.

Импорт проекта

Если имеется проект Eclipse (такой проект имеет файл `.project` в корне), то производится импорт существующего проекта:

1. Из меню выберите пункт **File -> Import...** Откроется выбор мастеров импорта.
2. Выберите категорию **General -> Existing Project into Workspace** и нажмите **Next**.
3. Выберите директорию проекта.
4. Затем выберите проекты для импорта.
5. Закончите импорт нажатием на **Finish**.

В случае, если присутствует только структура проекта `pgCodeKeeper`, без файлов проекта Eclipse, то импорт производится с помощью мастера импорта. Хранение структуры проекта `pgCodeKeeper` без информации Eclipse – обычная практика, например при использовании системы контроля версий.

1. Из меню выберите пункт **File -> Import ...** Откроется выбор мастеров импорта.
2. Выберите **pgCodeKeeper Project** в категории `pgCodeKeeper` и нажмите **Next**.

3. Укажите путь до директории со структурой проекта pgCodeKeeper, введите имя проекта и выберите его кодировку.
4. Выбираем тип базы данных для импортируемого проекта **Database type** / **Тип базы данных**.
5. Для добавления проекта в рабочую область, можно выбрать соответствующую опцию.
6. Нажмите **Finish**.

Для корректной работы проект должен содержать файл „pgcodekeeper“ в корне директории. Мастер импорта проекта автоматически создаст данный файл при его отсутствии.

Структура проекта

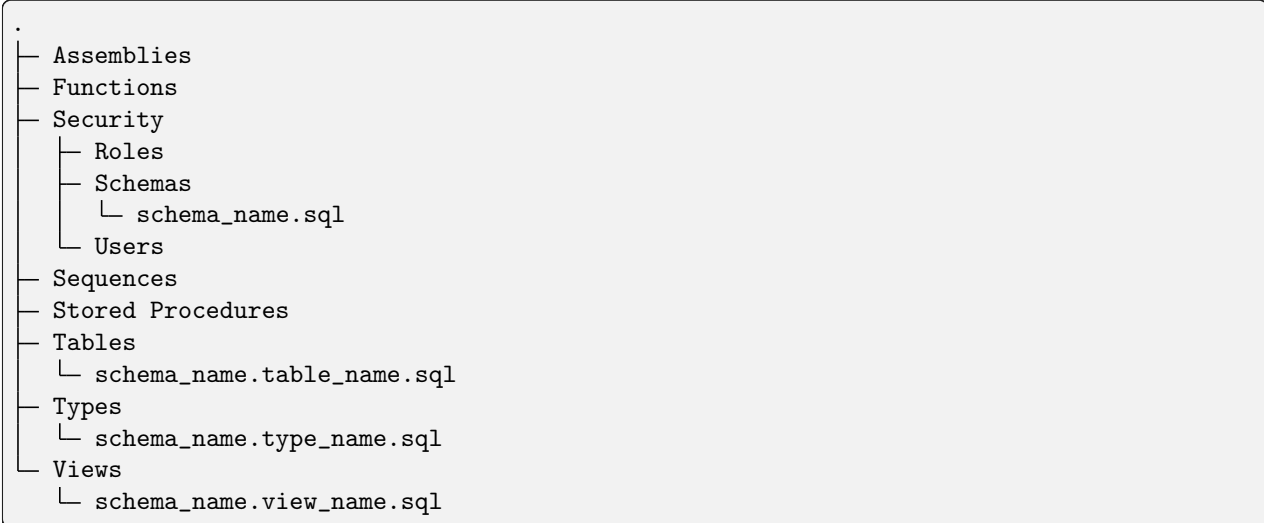
Проект **PostgreSQL** имеет следующую структуру:

```
.
├── EXTENSION
├── SCHEMA
│   ├── schema_name
│   ├── ...
│   └── another_schema_name
│       ├── another_schema_name.sql
│       ├── DOMAIN
│       ├── FTS_CONFIGURATION
│       ├── FTS_DICTIONARY
│       ├── FTS_PARSER
│       ├── FTS_TEMPLATE
│       ├── FUNCTION
│       ├── OPERATOR
│       ├── PROCEDURE
│       ├── SEQUENCE
│       ├── TABLE
│       │   └── table_name.sql
│       ├── TYPE
│       │   └── type_name.sql
│       └── VIEW
│           └── view_name.sql
```

- `schema_name ... another_schema_name` - имена схем.
- `another_schema_name.sql` - обязательный файл с определением схемы, имя должно совпадать с именем директории.
- `table_name.sql`, `type_name.sql`, `view_name.sql` - файлы с определениями объектов. Имена указываются без имени схем.

- все перегруженные функции хранятся в одном файле. Например, функции **public.func(int)** и **public.func(text)** будут расположены в **SCHEMA/public/FUNCTION/func.sql**.
- определения индексов, ограничений, триггеров и правил расположены в файлах родительских объектов. Например, индекс **public.ii** для таблицы **public.t1** будет расположен в **SCHEMA/public/TABLE/t1.sql**.

Проект **MS SQL** содержит следующую структуру:



- **schema_name.sql** - обязательный файл с определением схемы.
- **schema_name.table_name.sql**, **schema_name.type_name.sql**, **schema_name.view_name.sql** - файлы с определениями объектов. Имена указываются с именем схемы.
- определения индексов, ограничений, триггеров расположены в файлах родительских объектов. Например, индекс **dbo.ii** для таблицы **dbo.t1** будет расположен в **Tables/dbo.t1.sql**.

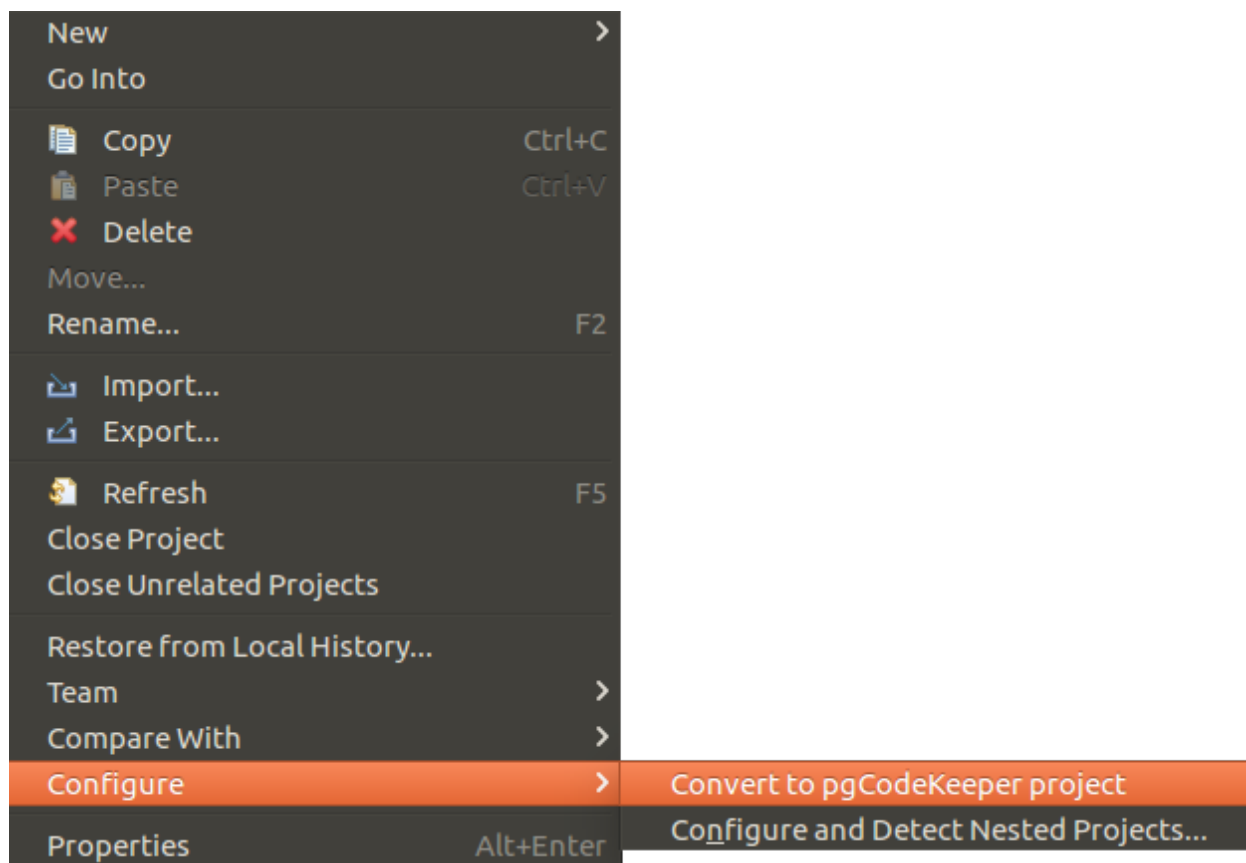
Кроме этого в проекте могут содержаться следующие директории:

MIGRATION. Содержит сохраненные скрипты миграции. Автоматическое сохранение и удаление скриптов миграции в данной директории можно выбрать на странице настроек [Обновление БД](#).

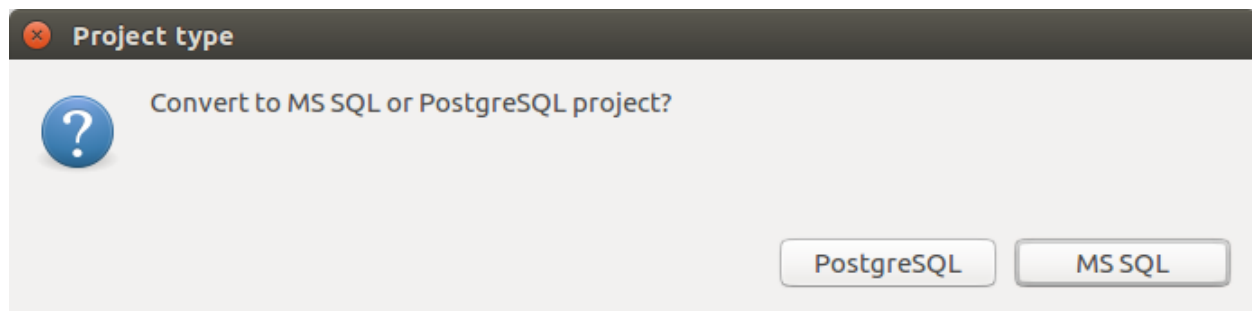
OVERRIDES. Содержит файлы переопределения свойств объектов. Внутренняя структура данной директории дублирует структуру проекта. Сохранить переопределения свойств возможно выбрав соответствующую опцию при сохранении объектов в проект. На данный момент поддерживается переопределение привилегий и владельцев объектов базы данных.

Конвертация проекта

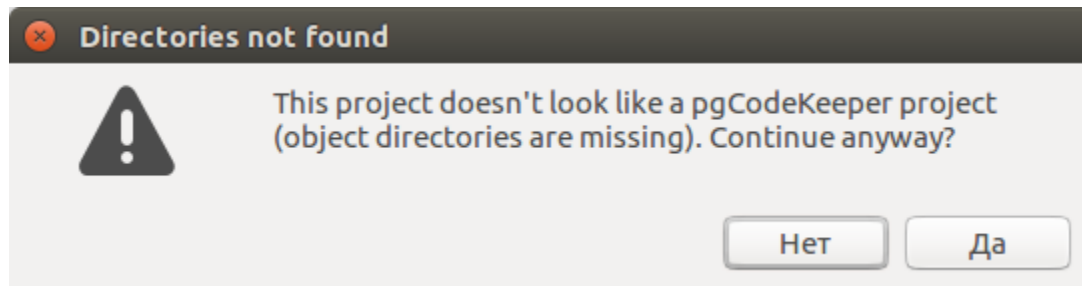
Любой проект можно сконвертировать в проект pgCodeKeeper. Для этого из меню проекта выберите пункт **Configure -> Convert to pgCodeKeeper project**.



В некоторых случаях может потребоваться выбрать тип проекта: PostgreSQL или MS SQL.



Если структура проекта не соответствует стандарту, будет показано предупреждение.

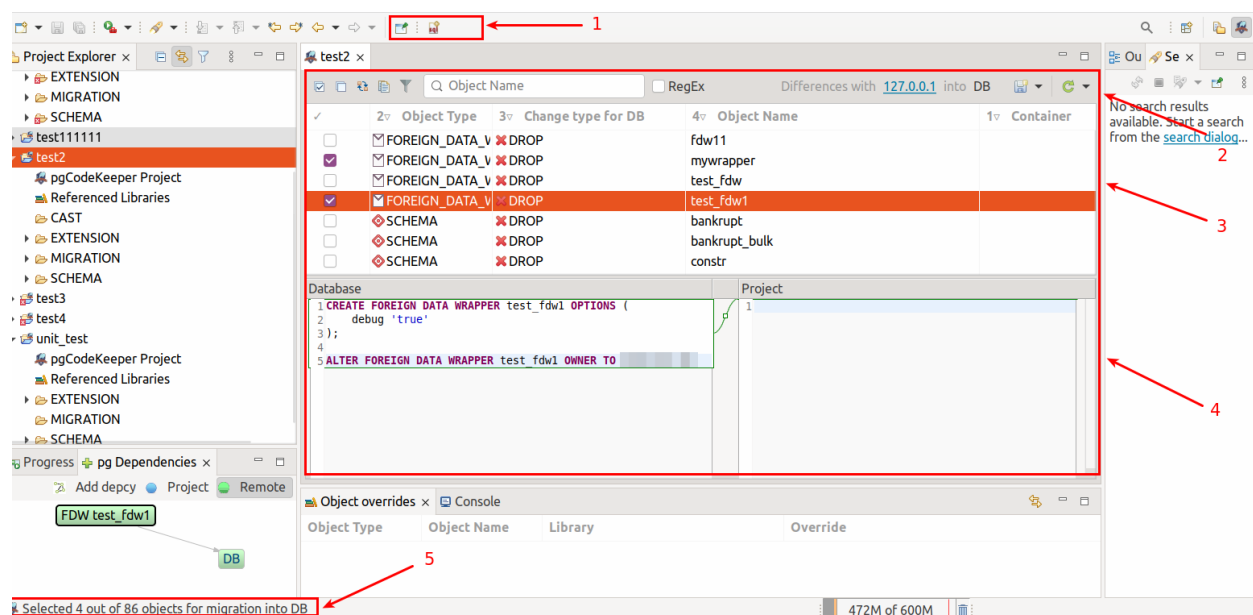


Редакторы и панели

7.1 Редактор проекта pgCodeKeeper

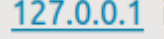
Редактор проекта – основной способ работы с pgCodeKeeper. Интерфейс редактора, состоящий из нескольких основных частей:

1. Панель инструментов Eclipse
2. Панель инструментов редактора
3. Таблица различий
4. Панель сравнения
5. Счетчик объектов



Панель инструментов Eclipse содержит инструменты для работы с редактором SQL. Кнопка  позволяет открыть новый *SQL редактор*.

Панель инструментов редактора предназначена для работы со списком объектов. Существуют следующие возможности:

-  – выбрать все элементы.
-  – снять выбор со всех элементов.
-  – инвертировать выбор элементов.
-  – копировать набор выбранных элементов в буфер обмена как регулярное выражение.
-  – неактивный /  – активный фильтр объектов.
-  – отфильтровать список объектов по имени или регулярному выражению.
-  – применить изменения
- ▼ – выпадающее меню с направлениями изменений. При клике по лейблу DB/БД также отображается меню с направлениями изменений.
-  – получить изменения
- ▼ – выпадающее меню с источниками баз данных. Источники баз данных можно выбрать в контекстном меню выбранной БД  на панели инструментов редактора (2).

Выпадающие меню ▼ позволяют запустить операции получения или применения изменений с перепределением некоторых настроек.

Таблица различий отображает список объектов, которые различаются в сравниваемых схемах БД. Здесь доступна информация: тип объекта, тип изменения, имя объекта, контейнер, пользователь git, пользователь базы данных.

Тип объекта - поддерживаются следующие типы объектов: SCHEMA, TYPE, SEQUENCE, TABLE, FUNCTION, PROCEDURE, VIEW, CONSTRAINT, INDEX, TRIGGER.

Для **PostgreSQL** дополнительно поддерживаются следующие типы: CAST, EXTENSION, DOMAIN, OPERATOR, FTS_PARSER, FTS_TEMPLATE, FTS_DICTIONARY, FTS_CONFIGURATION, AGGREGATE, RULE, POLICY, EVENT TRIGGER.

Для **MS SQL** дополнительно поддерживаются следующие типы: USER, ROLE, ASSEMBLY.

Тип изменения - объект может находиться в трех состояниях: существовать только в базе данных, только в проекте или и в проекте и в базе данных. В зависимости от состояния и направления изменений, объекты в списке изменений помечаются как delete, add и edit или CREATE, DROP и ALTER соответственно.

Контейнер - имя родительского объекта, например для индекса - это имя таблицы или представления, которому он принадлежит.

Пользователь git - имя пользователя, который последним изменил файл проекта для данного объекта. В том случае, если файлы были изменены локально, то к имени пользователя будет добавлен символ „*“. Для отображения данного столбца, проект должен быть подключен к системе контроля версий и должна быть включена соответствующая опция на странице настроек *Редактор проекта*.

Пользователь базы данных - имя пользователя базы данных, который последним изменил объект в базе данных. Для отображения данного столбца необходимо расширение `pg_dbo_timestamp`.

Панель сравнения показывает изменения, произошедшие в SQL представлении объекта.

Счетчик объектов отображает выбранное и общее количество объектов в таблице.

7.2 Поиск объекта в таблице различий

Для поиска объектов в таблице различий удобно воспользоваться полем поиска над списком изменений.

Поиск происходит по именам объектов. При установленном флажке **RegEx** поиск осуществляется с использованием регулярных выражений.

Для начала просто введите часть имени в поле поиска. В таблице различий отобразятся объекты, соответствующие данному условию.

Поле поиска хранит последние 200 запросов.

Для поиска по квалифицированному имени объекта введите в поле поиска имя схемы с ..

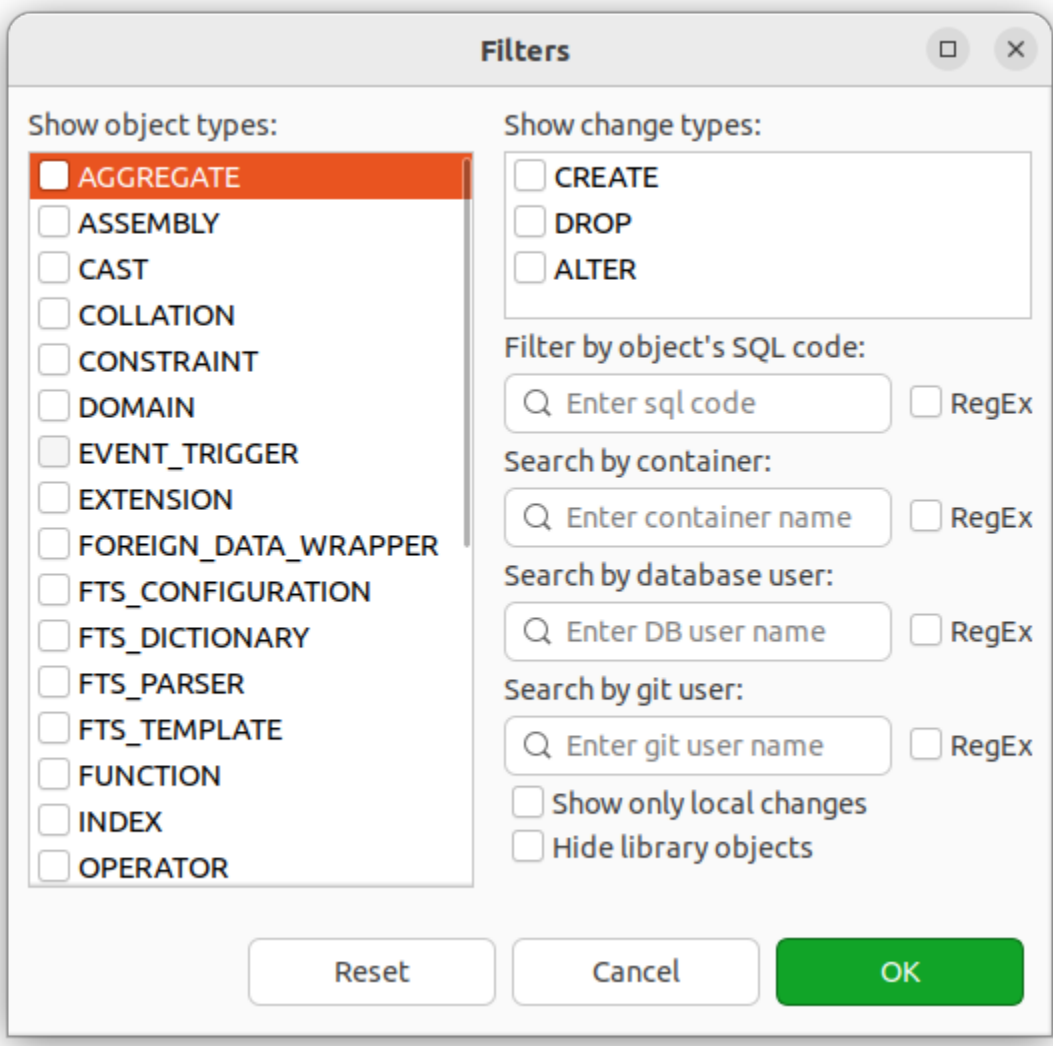
✓	2▼ Object Type	3▼ Change type for DB	4▼ Object Name	1▼ Container
☐	FUNCTION	+ CREATE	get_city()	country
☐	TABLE	+ CREATE	city	country
☐	TYPE	+ CREATE	towns	country

В результате в таблице различий отобразятся все объекты из этой схемы, кроме самой схемы.

Для отображения только схемы введите его имя.

7.3 Фильтры объектов

Для фильтрации объектов можно использовать диалоговое окно, которой отобразится при нажатии кнопки  над списком изменений.



Список типов объектов позволяет отобразить только выбранные типы объектов. Если выбран тип TABLE или VIEW будут отображены все дочерние объекты.

Список типов изменений позволяет отобразить только выбранные типы изменений.

Filter by object's SQL code / Фильтр по SQL коду объекта ищет в совпадения коде генерации объектов.

Search by container / Поиск по контейнеру ищет объекты, находящиеся в схемах с подходящими именами. Также будут отображены и сами объекты схем.

Search by database user / Поиск по пользователю БД ищет совпадения по авторам изменений объектов в сторонней базе данных. Для использования необходимо расширение `pg_dbo_timestamp`.

Search by git user / Поиск по пользователю git ищет совпадения по авторам последнего изменений объекта в локальном репозитории. Для использования проект должен находиться под системой контроля версий.

Show only local changes / Показывать только локальные изменения отображает изменения, произошедшие в локальном репозитории с момента последнего коммита. Для использования проект должен находиться под системой контроля версий.

Hide library objects / Скрывать библиотечные объекты скрывает объекты, загруженные из библиотек.

Для применения фильтров нажмите кнопку **OK**.

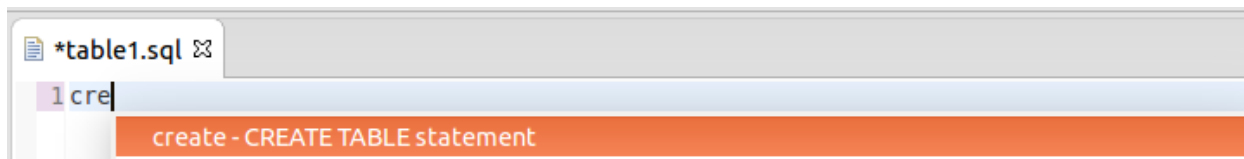
Для отмены текущих изменений нажмите кнопку **Cancel** / **Отмена**.

Для сброса всех фильтров нажмите кнопку **Reset** / **Сбросить**.

При включенном фильтре иконка кнопки изменится на .

7.4 SQL редактор

SQL редактор внешне не отличается от обычного текстового редактора Eclipse, но в нем имеется возможность подсветки синтаксиса SQL, автоподстановки шаблонов SQL запросов, навигации по объектам схемы базы данных и форматирования кода.



При работе с SQL редактором в **панели инструментов Eclipse** доступны следующие возможности:

 - Quick update / Быстрое обновление. Исполнение скрипта миграции для **различий** между объектом в текущем файле проекта и соответствующим объектом в базе данных. Если изменения затрагивают объекты из других файлов или модифицируют данные (например удаление колонки), то обновление не произойдет.

 - Execute selection / Выполнить выбранное. Исполнение текущего скрипта миграции (или выделенного текста) в выбранной базе данных.

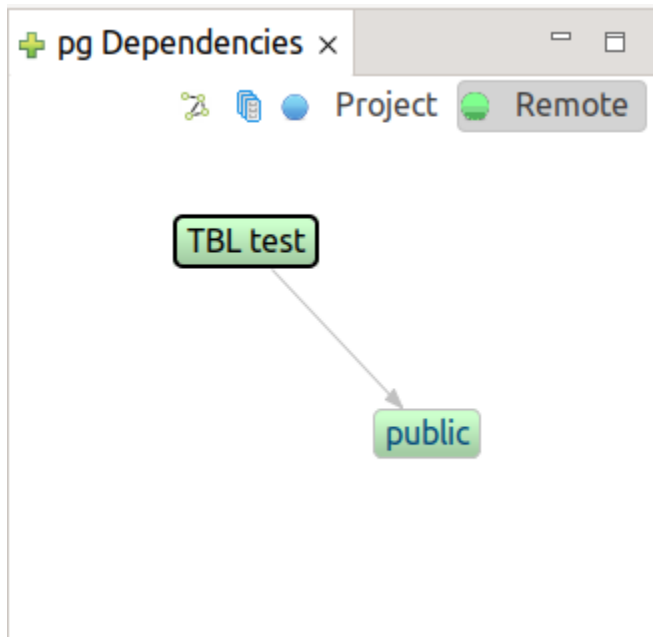
 - Cancel execution / Отменить выполнение. Отмена исполнения текущего скрипта миграции.

 - Get changes / Получить изменения. Обновляет редактор проекта, в котором находится текущий файл. Применимо только к файлам проекта.

Для форматирования выделенного участка кода нажмите горячую клавишу Shift + Ctrl + F или в контекстном меню выберите **Format** / **Форматировать**.

7.5 Просмотр зависимостей объекта БД

Панель **pg Dependencies** / **pg Зависимости** показывает зависимости, в которых участвует текущий объект, выбранный в панели различий активного редактора проекта.



Стрелки указывают на направление связи от зависящего объекта к его зависимости.

Переключатель **Project - Remote** / **Проект - БД** позволяет выбрать, для какой БД, участвующей в сравнении, показывать объекты и зависимости. После переключения необходимо повторно выбрать элемент в панели различий редактора.

Кнопка  **Show columns** / **Показать колонки** позволяет отобразить колонки таблиц текущего объекта и его зависимостей.

Кнопка  **Add depcy** / **Добавить зависимости** позволяет открыть диалог ручного добавления зависимостей.

Set additional dependencies...

Database

Object:

Depends on:

Add →

Dependant → Dependency

public.f() → public.tt

Remove

Project

Object:

Depends on:

Add →

Dependant → Dependency

Remove

?

Cancel OK

В этом диалоге можно явно задать зависимости между объектами БД. Это может понадобиться, например, в случае если автоматическое распознавание зависимостей не сработает для некоторой сложной зависимости. Добавленные зависимости будут учтены при генерации последовательности выражений скрипта наката.

Окно состоит из двух частей, которые служат для добавления зависимостей к сравниваемым БД.

Для добавления зависимостей между объектами, достаточно ввести начальные буквы из названия объекта и из выпадающего списка выбрать имена зависимых объектов и нажать на кнопку **Add / Добавить**. Зависимость отобразится в списке добавленных.

Для удаления выделите связку зависимых объектов и нажмите **Remove / Удалить**.

7.6 Переопределенные объекты

Панель **Object overrides** / **Переопределенные объекты** отображает список переопределенных объектов в библиотеках для активного редактора проекта. Данный вид автоматически отображается при получении изменений, если присутствуют хотя бы один конфликт.

Object Type	Object Name	Library	Override
SCHEMA	public	/home/galiev_mr/1.sql	/home/galiev_mr/runtime-Config/test project/SCHEMA/public/public.sql
TABLE	t1	/home/galiev_mr/1.sql	/home/galiev_mr/runtime-Config/test project/SCHEMA/public/TABLE/t1.sql

Кнопка  отображает записи, относящиеся только к объектам, выбранным в панели различий активного редактора проекта.

Контекстное меню позволяет открыть обе версии объекта, а также просмотреть различия в специальном редакторе сравнения.

7.7 Результат запроса

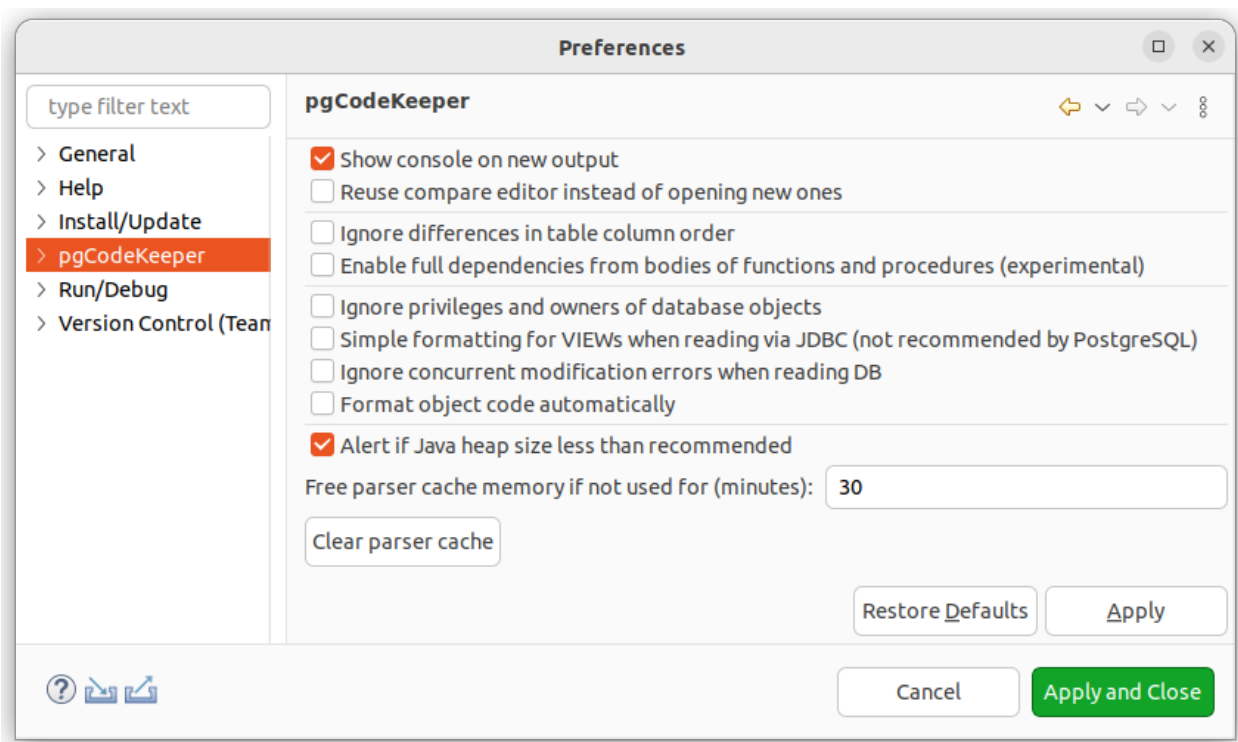
Панель **Query result** / **Результат запроса** отображает результат запросов. Каждый запрос отображается отдельной вкладкой.

Внимание: Большие выборки могут вызывать «подвисания» графического интерфейса.

Query result				
select * from t1				
	c1 integer	c2 integer	c3 integer	c4 text
1	1	2	3	<NULL>
2	4	2	3	<NULL>
3	4	1	3	<NULL>
4	1	2	<NULL>	<NULL>

2019-10-01 16'36'41
2019-10-01 16'37'05

8.1 Основные настройки pgCodeKeeper



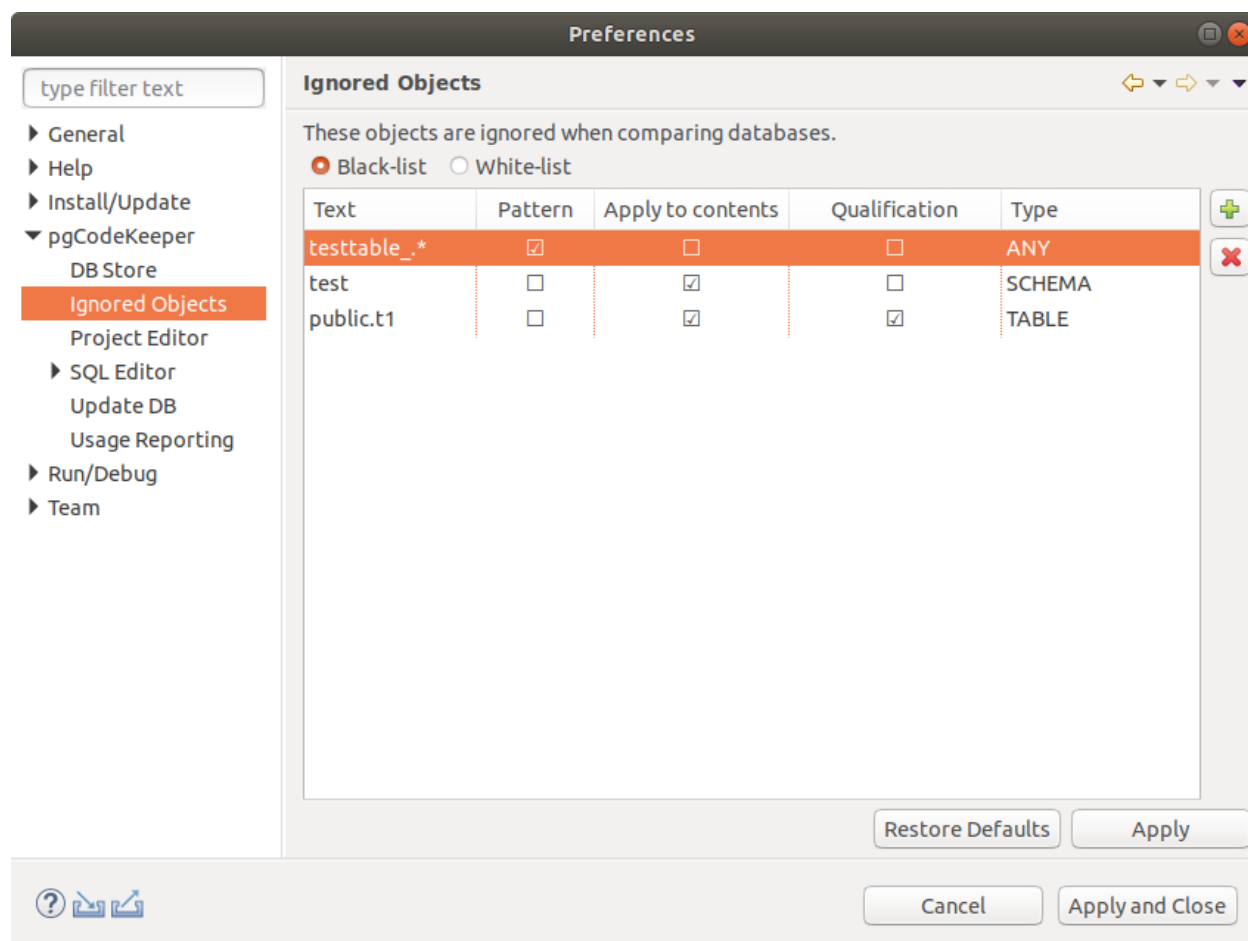
- **Show console on new output / Показывать консоль** - позволяет автоматически показывать консоль программы при выводе в нее.
- **Ignore privileges and owners of database objects / Не учитывать привилегии и владельцев объектов** - позволяет отключить поиск различий в свойствах объектов, связанных с

ролями БД.

- **Ignore differences in table column order / Игнорировать различия в порядке столбцов таблицы** - позволяет не учитывать порядок столбцов при сравнении таблиц.
- **Reuse compare editor instead of opening new ones / Использовать открытый редактор сравнения вместо создания новых** - позволяет использовать открытый редактор сравнения при сравнения объектов базы данных (В контекстном меню таблицы различий выбрать пункт **Show Diff / Показать различия**).
- **Enable full dependencies from bodies of functions and procedures (experimental) / Учитывать все зависимости из тел функций и процедур (экспериментально)** - позволяет искать внутри тел функций и процедур зависимости к другим функциям и процедурам.
- **Simple formatting for VIEWS when reading via JDBC (not recommended by PostgreSQL) / Упрощенное форматирование представлений при чтении через JDBC (не рекомендовано PostgreSQL)** - позволяет использовать упрощенное форматирование представлений, убирая лишние скобки в выражениях. Данный формат может не поддерживаться в будущих версиях PostgreSQL.
- **Ignore concurrent modification errors when reading DB / Игнорировать ошибки конкурентной модификации объектов при чтении БД** - позволяет игнорировать ошибки, возникающие при модификации объекта БД во время его чтения.
- **Format object code automatically / Форматировать код объектов автоматически** - позволяет отображать и сохранять отформатированный код функций на языках plpgsql и sql согласно настройкам форматирования **SQL Editor/Редактор SQL**. Подробнее про настройки форматирования кода в *Форматирование*.
- **Free parser cache memory if not used for (minutes): / Освобождать память кэша парсера если он не использовался последние (минут):** - позволяет автоматически очищать неиспользуемый кэш парсера через определенный промежуток времени. Значение равное 0 отключает данное поведение.
- **Clear parser cache / Очистить кэш парсера** - позволяет очистить кэш парсера.

8.2 Исключенные объекты

Объекты, которые не должны учитываться при сравнении БД, возможно указать на странице настроек **pgCodeKeeper -> Ignored Objects / Исключенные объекты**.



Переключатель **Черный список - Белый список** позволяет инвертировать черный список в белый и обратно.

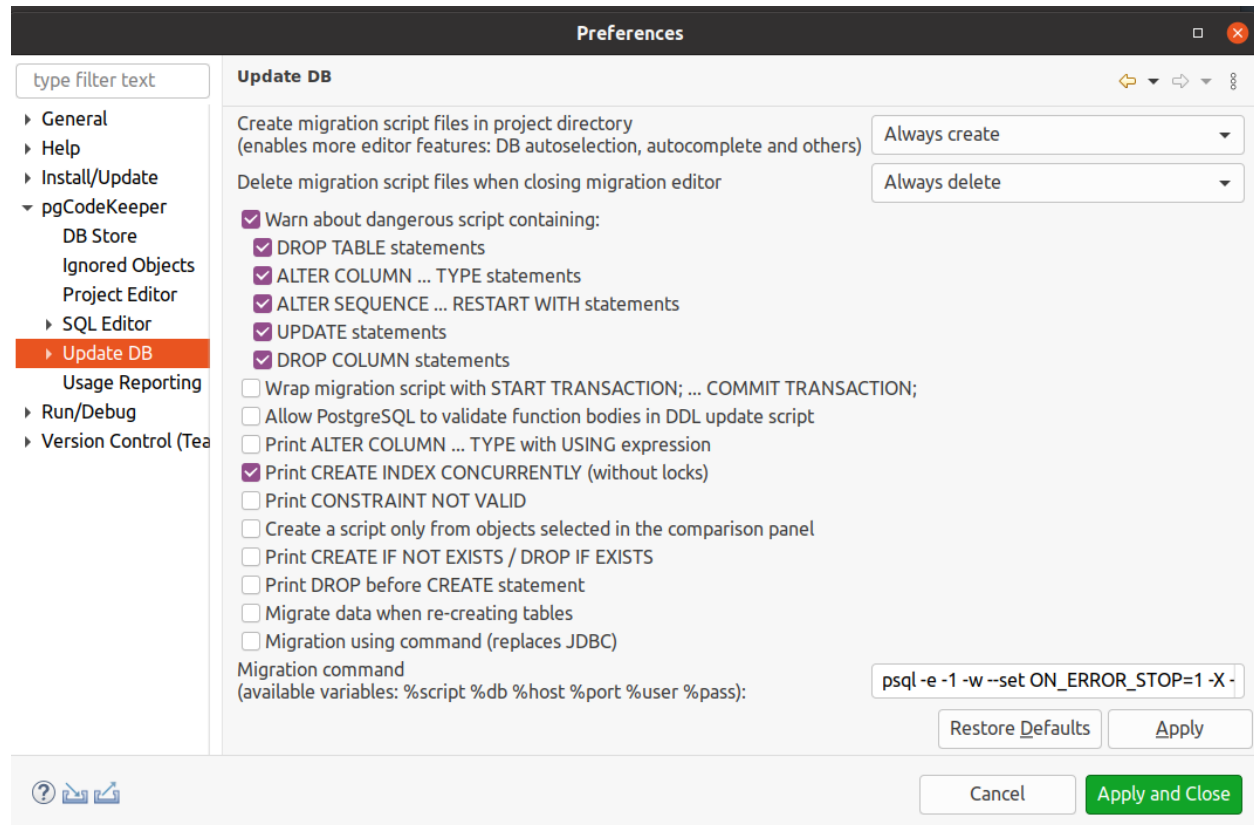
- **Text / Текст** – имя объекта.
- **Pattern / Шаблон** – воспринимать имя объекта как регулярное выражение.
- **Apply to contents / Применить к содержимому** – применять правило для совпавшего объекта и всего его содержимого.
- **Qualification / Квалификация** – искать объекты по квалифицированному имени.
- **Type / Тип** – тип объекта («ANY» предназначен для указания любого типа объекта).

Для добавления объекта в список исключенных или удаления из него, служат кнопки  и  соответственно.

Примечание: В данном случае список исключаемых объектов будет **общим** и будет распространять свое действие на все проекты находящиеся в директории проектов. Подробное описание работы со списками находится в разделе [Список исключаемых объектов](#).

8.3 Обновление БД

Настройки, управляющие обновлением базы данных.

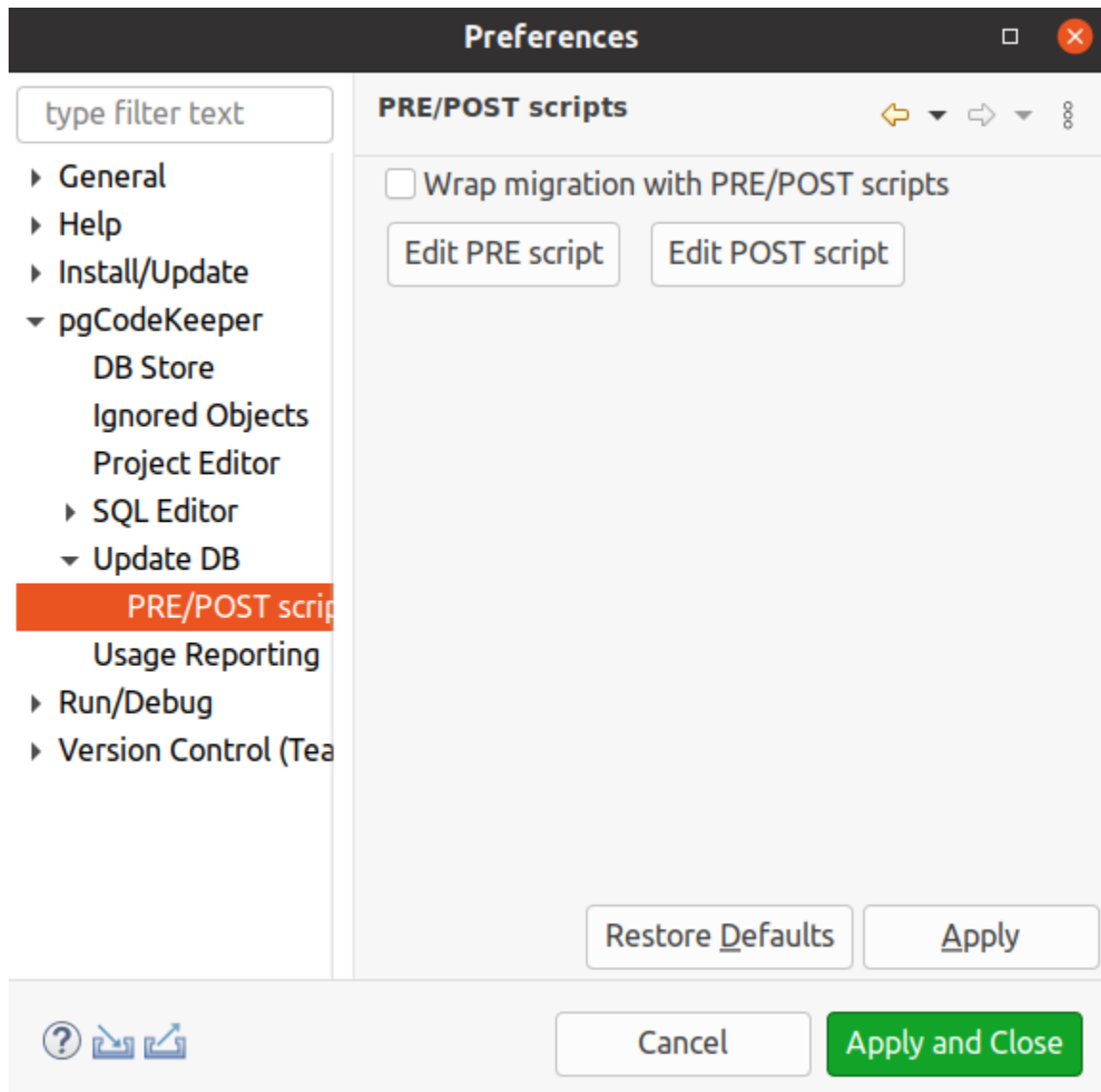


На странице настроек **pgCodeKeeper -> Update DB / Обновление БД** Вы можете настроить параметры для работы редактора скрипта наката.

- **Create migration script files in project directory / Создавать скрипт миграции в директории проекта** – определяет поведение при создании скриптов миграции.
- **Delete migration script files when closing migration editor / Удалять файлы скриптов миграции после закрытия редактора** – определяет поведение при закрытии скриптов миграции, если они были созданы в директории проекта.
- **Warn about dangerous scripts containing: / Предупреждать, если в скрипте присутствуют следующие выражения:** – позволяет предупреждать, если при генерации скрипта образовались выбранные опасные выражения.
- **Show script output in separate window / Показывать вывод скрипт наката в отдельном окне** – позволяет просмотреть сообщение с ответом сервера при выполнении наката.
- **Surround migration script with START TRANSACTION; ... COMMIT TRANSACTION; / Заключить скрипт наката в блок START TRANSACTION; ... COMMIT TRANSACTION;** – позволяет дополнять генерируемые скрипты парой start ... commit.
- **Allow PostgreSQL to validate function bodies in DDL update script / Разрешить PostgreSQL проверять тела функций в скрипте наката** – позволяет при генерации скрипта добавлять проверку `SET check_function_bodies = true;` в начало скрипта.

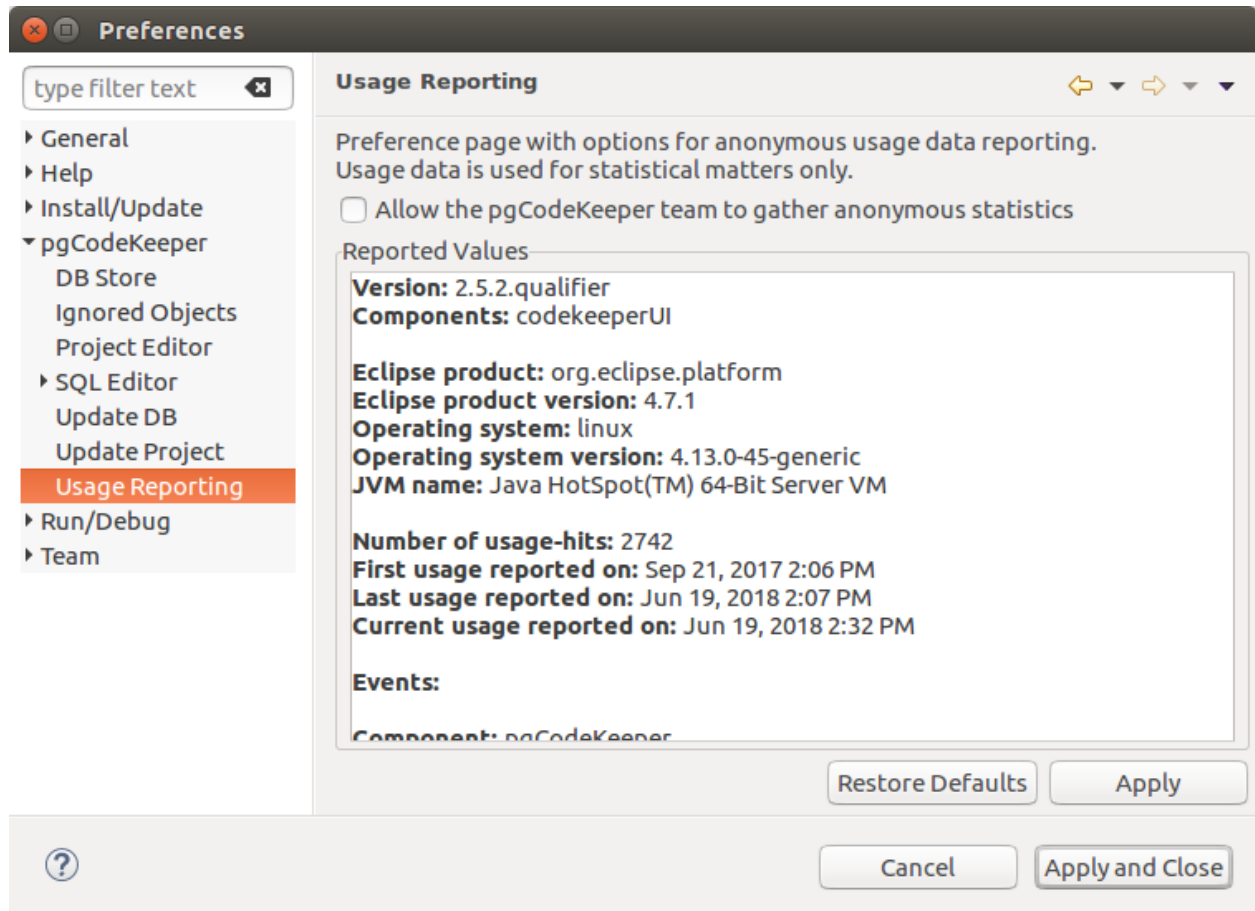
- **Print ALTER COLUMN ... TYPE with USING expression / Печатать ALTER COLUMN ... TYPE ... с USING expression** – позволяет добавлять в скрипт выражение, позволяющее изменить тип данных, при изменении типа столбца
- **Print CREATE INDEX CONCURRENTLY (without locks) / Печатать CREATE INDEX CONCURRENTLY (без блокировок)** – позволяет в скриптах создавать индексы в режиме CONCURRENTLY.
- **Print CONSTRAINT NOT VALID / Печатать CONSTRAINT NOT VALID** - позволяет добавлять в скрипт миграции конструкцию NOT VALID для констрейнтов.
- **Print CREATE IF NOT EXISTS/DROP IF EXISTS / Печатать CREATE IF NOT EXISTS/DROP IF EXISTS** – позволяет добавить конструкцию IF NOT EXISTS/IF EXISTS в скрипте миграции в конструкции CREATE / DROP объекта.
- **Print DROP before CREATE statement / Печатать DROP перед CREATE конструкции** – позволяет добавить конструкцию DROP перед CREATE конструкцией в скрипте миграции.
- **Create a script only from objects selected in the comparison panel / Создавать скрипт только из объектов, выбранных в панели сравнения** – позволяет исключить из скрипта объекты, которые не были выбраны явно.
- **Migrate data when re-creating tables / Мигрировать данные при пересоздании таблиц** - позволяет сохранить данные при пересоздании таблицы. При пересоздании существующая таблица переименовывается и создается новая, данные из исходной переливаются в новую. Старая переименованная таблица удаляется. Состояние IDENTITY SEQUENCE переносится из старой таблицы в новую. Обычные SEQUENCE не поддерживаются.
- **Migration command / Команда миграции** - позволяет использовать другую утилиту или параметры запуска при применении скрипта наката к базе данных.

Во вкладке **Update DB / Обновление БД** -> **PRE/POST script** есть настройки для добавления PRE/POST скриптов для миграций. В проекте возможно создать директории PRE и POST, содержимое которых будет добавлено в начало и конец основного скрипта. Также доступны PRE/POST скрипты в глобальных настройках и в опциях CLI. Для редактирования содержимого скриптов PRE/POST выберите соответствующие кнопки.



8.4 Отчеты об использовании

Настройки, управляющие отправкой отчетов.

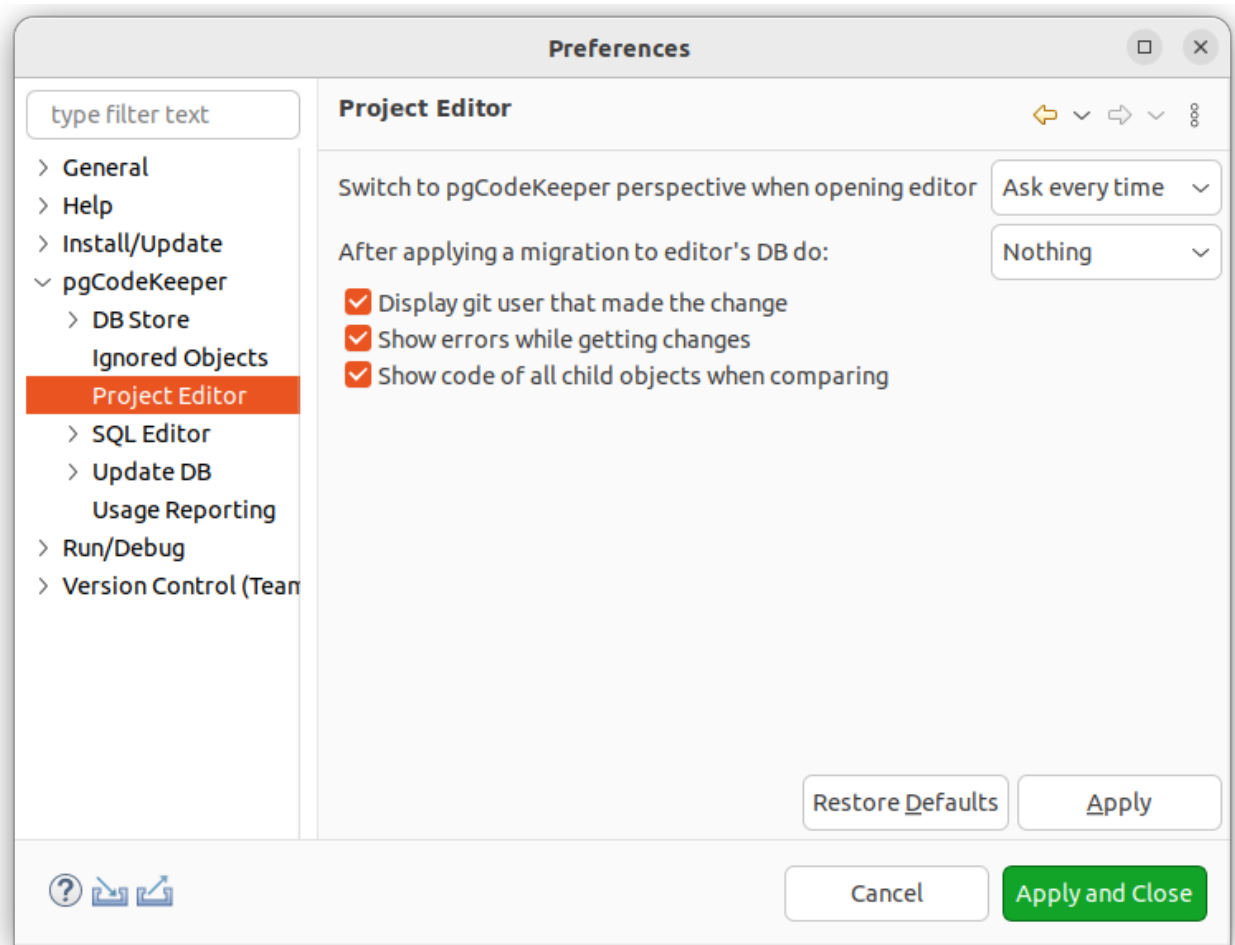


На странице настроек **pgCodeKeeper** -> **Usage Reporting** / **Отчеты об использовании** Вы можете включить или выключить отправку статистики об использовании **pgCodeKeeper**.

Отправляемые значения для статистики Вы можете просмотреть в поле **Reported Values** / **Отправляемые значения**.

8.5 Редактор проекта

Настройки, управляющие редактором проекта.

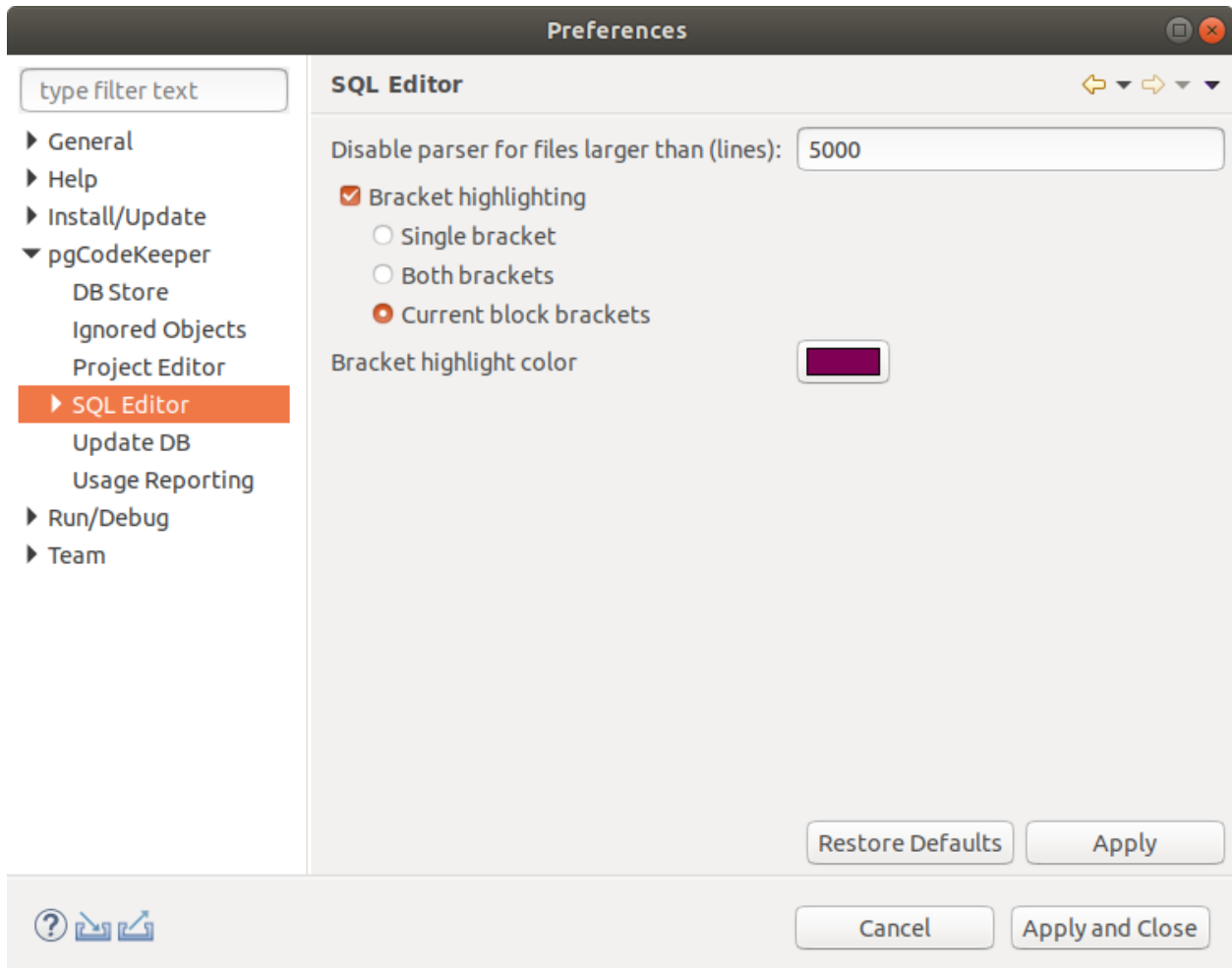


На странице настроек **pgCodeKeeper -> Project editor / Редактор проекта** Вы можете настроить параметры редактора проекта.

- **Switch to pgCodeKeeper perspective when opening editor / Переключать в перспективу pgCodeKeeper при открытии редактора** – определяет поведение переключения перспективы при открытии редактора проекта.
- **After applying a migration to editor's DB do: / После выполнения миграции на БД редактора выполнить:** – определяет поведение редактора при накате скриптов миграции.
- **Display git user that made the change / Отображать пользователя git, совершившего изменение** – позволяет отображать в редакторе колонку с пользователем git, совершившем последнее изменение файла. Проект должен находиться под системой контроля версий.
- **Show errors while getting changes / Показывать ошибки во время получения изменений** – позволяет отображать список ошибок, найденных во время получения изменений.
- **Show code of all child objects when comparing / Показывать код всех дочерних объектов при сравнении** - позволяет отображать в панели сравнения код всех дочерних объектов даже если в них нет различий.

8.6 Редактор SQL

Настройки, управляющие редактором SQL.



На странице настроек **pgCodeKeeper -> SQL Editor / Редактор SQL** Вы можете настроить параметры редактора SQL.

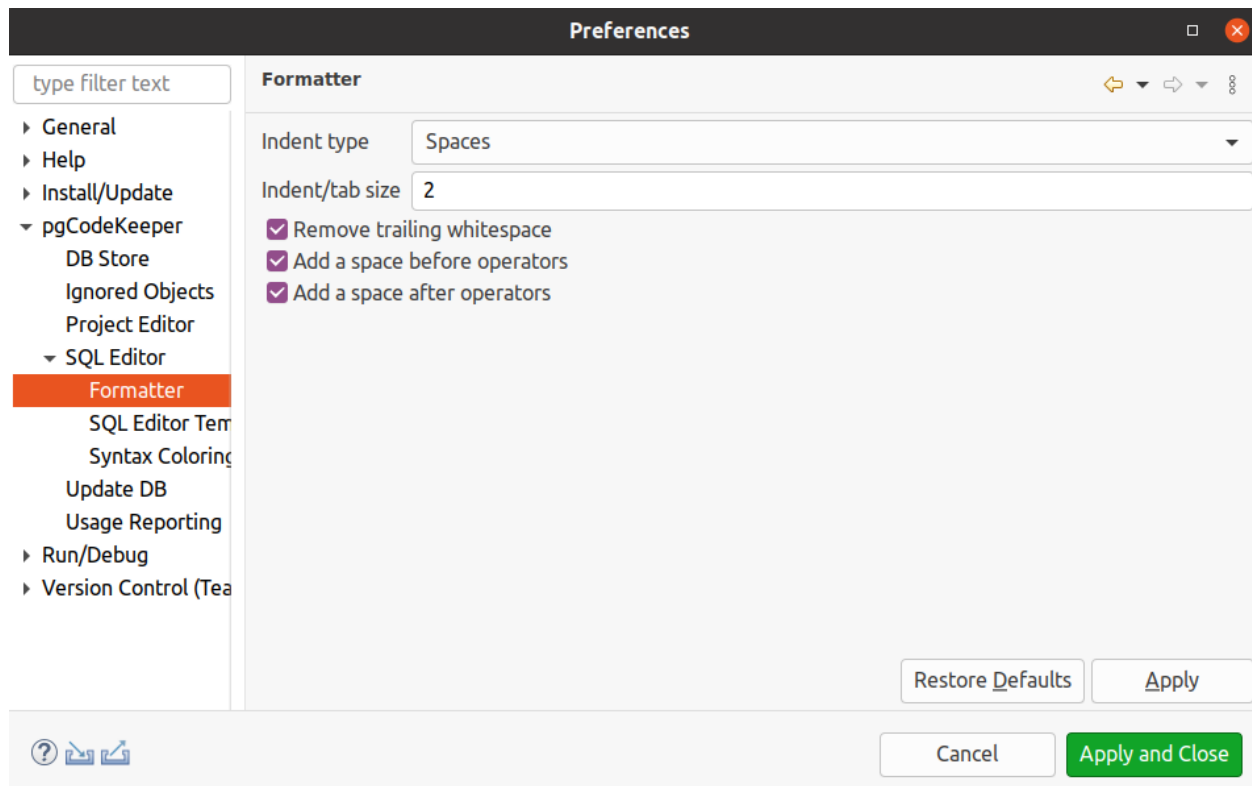
- **Disable parser for files larger than (lines): / Отключить парсер для файлов больше чем (строки):** – позволяет выбрать размер файла, для которого отключается сборщик pgCodeKeeper. Это ускоряет открытие и работу с файлом, но отключает поиск ошибок и ссылок на объекты внутри этого файла. Значение равное 0 отключает данное поведение.
- **Bracket highlight / Подсветка скобок** – включает подсветку скобок в одном из режимов.
- **Single bracket / Одна скобка** – при выборе скобки, подсвечивает парную скобку.
- **Both brackets / Обе скобки** – при выборе скобки, подсвечивает текущую и парную скобку.
- **Current block brackets / Скобки текущего блока** – подсвечивает скобки, между которыми находится курсор.
- **Bracket highlight color / Цвет подсветки скобок** – позволяет выбрать цвет подсветки скобок.

8.7 Форматирование

Настройки для управления правилами форматирования.

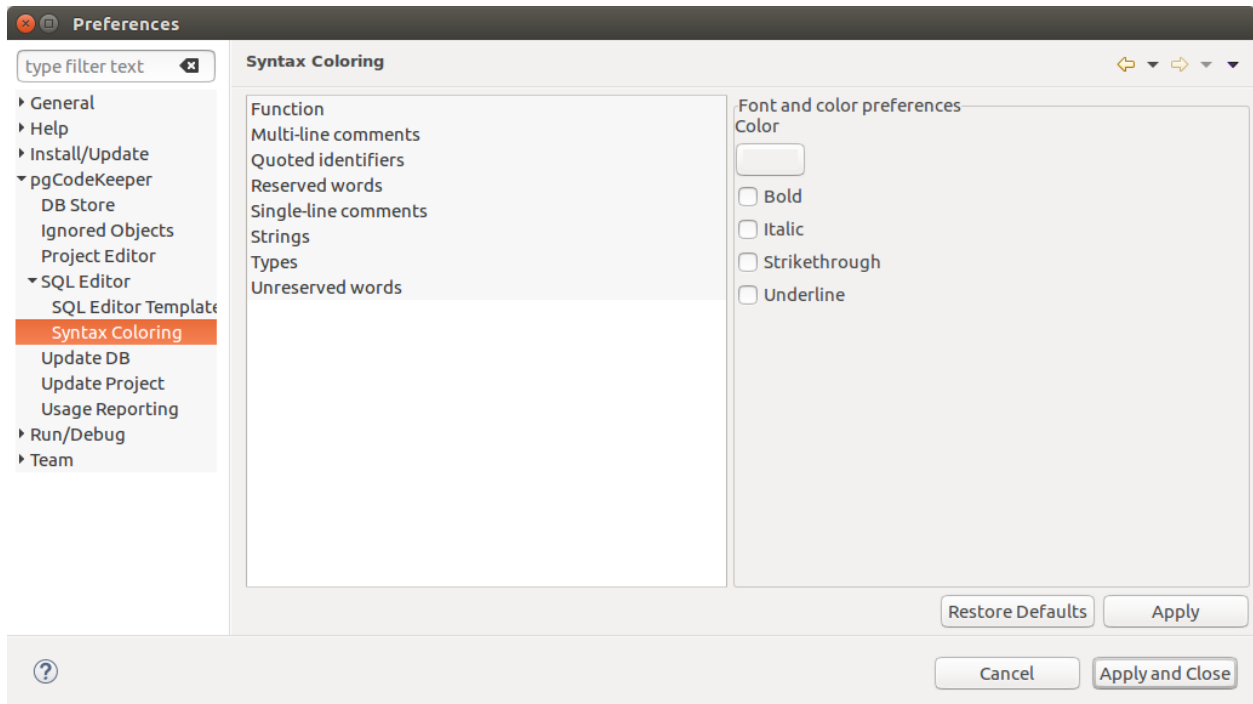
На странице настроек **pgCodeKeeper** -> **SQL Editor** / Редактор SQL -> **Formatting** / **Форматирование** можно установить:

- тип отступа, применяемый в начале строки кода;
- задать ширину отступа для выбранного типа отступа;
- включить/отключить настройку удаления пробелов в конце строки;
- добавление пробелов перед/после операторов.



8.8 Подсветка синтаксиса

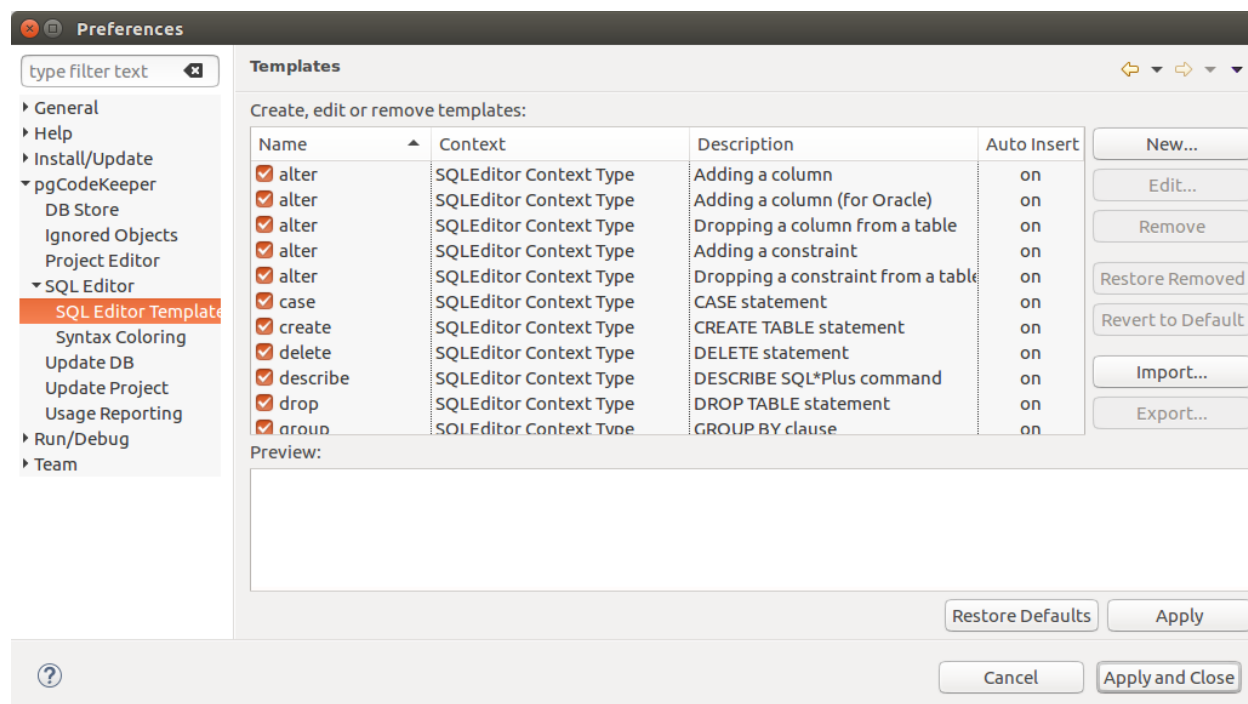
Настройки, управляющие подсветкой синтаксиса.



На странице настроек **pgCodeKeeper -> SQL Editor / Редактор SQL -> Syntax Coloring / Подсветка синтаксиса** Вы можете установить цвет и стиль шрифта, которым будет отображаться соответствующий синтаксис в SQL редакторе pgCodeKeeper.

8.9 Шаблоны редактора SQL

Создание, удаление и редактирование шаблонов для автоподстановки в SQL редакторе

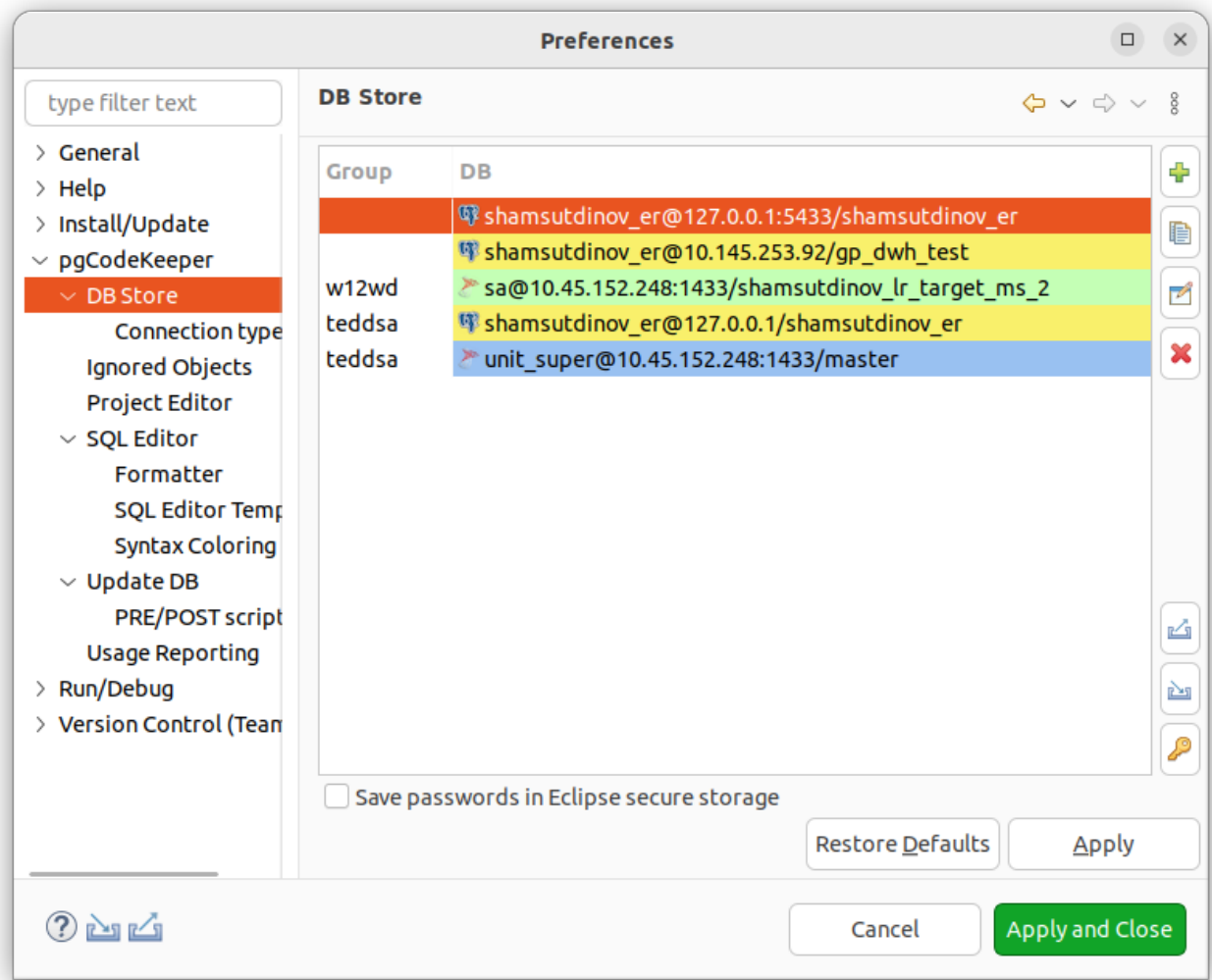


На странице настроек **pgCodeKeeper** -> **SQL Editor** / **Редактор SQL** -> **SQL Editor Templates** / **Шаблоны редактора SQL** Вы можете управлять SQL шаблонами, доступными в SQL редакторе **pgCodeKeeper**. Список доступных действий отображается справа от списка шаблонов.

8.10 Хранилище БД

Настройки, управляющие хранилищем параметров подключения к БД.

На странице настроек **pgCodeKeeper** -> **DB Store** / **Хранилище БД** находится список записей для подключения к базам данных. Для загрузки списка подключений из файла нажмите по кнопке  **Import connection list** / **Импортировать список подключений** и выберите необходимый файл с подключениями. Также вы можете сохранить список подключений в отдельный файл, для этого нажмите по кнопке  **Export connection list** / **Экспортировать список подключений**.



Для добавления хранилища нажать кнопку . Откроется диалог **DB credentials** (Данные подключения к БД), в котором нужно указать параметры данных подключения: хост, порт, имя БД, пользователь, пароль, группа БД, тип соединения, а также включить запрет на запись в БД и выбрать тип БД (PostgreSQL или MS SQL). Для MS SQL можно указать домен. Если выбран тип бд MS SQL, то по умолчанию выставлен параметр **trust MS SQL certificate** / **доверять сертификату MS SQL**, который можно отключить. Название записи можно указать вручную, для этого нужно отключить опцию **Auto-generate** / **Автогенерация**.

Примечание: pgCodeKeeper поддерживает работу с **pgpass** файлом. Для этого поле с паролем нужно оставить пустым.

Create new DB connection

DB Info Ignored objects files Connection properties External loader

DB Host: 127.0.0.1 Port: 5432

DB Name:

DB User:

DB Password:

.pgpass passwords are supported

Read only ☐ disallow executing scripts on DB

Database type PG

MS SQL certficiate ☒ trust MS SQL certificate

Domain:

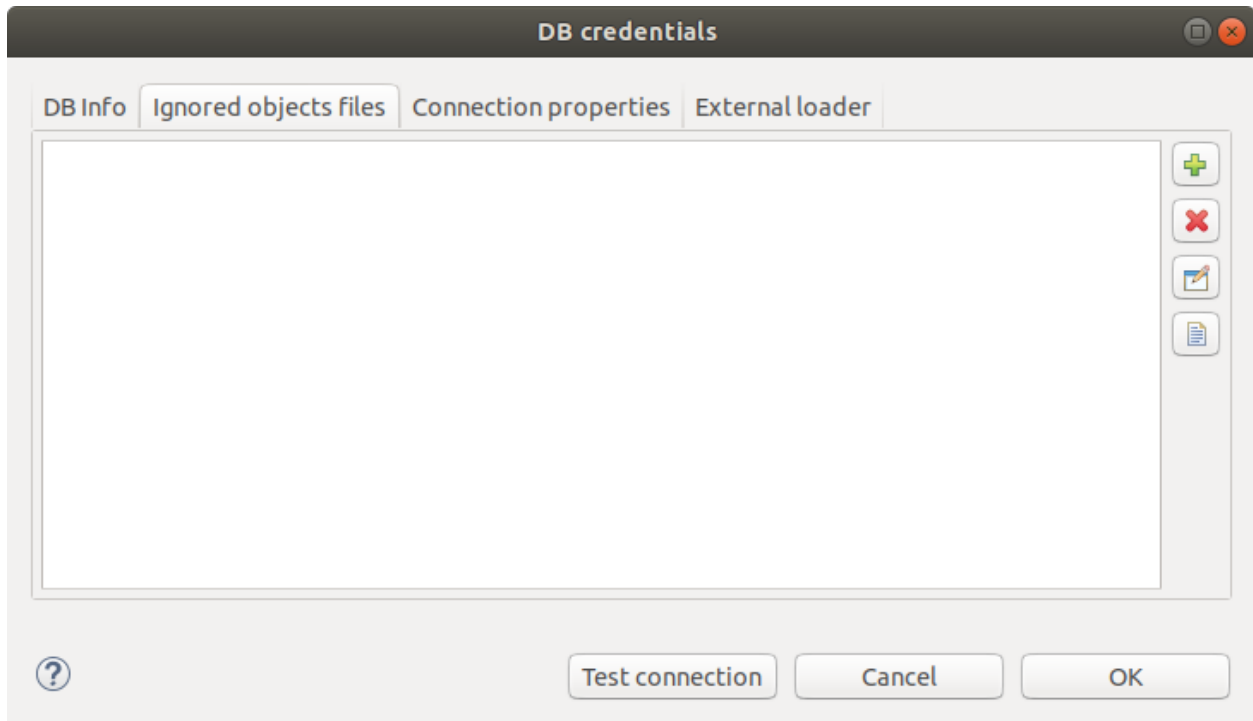
Entry Name: 127.0.0.1 ☒ Auto-generate

Group:

Connection Type

? Test connection Cancel OK

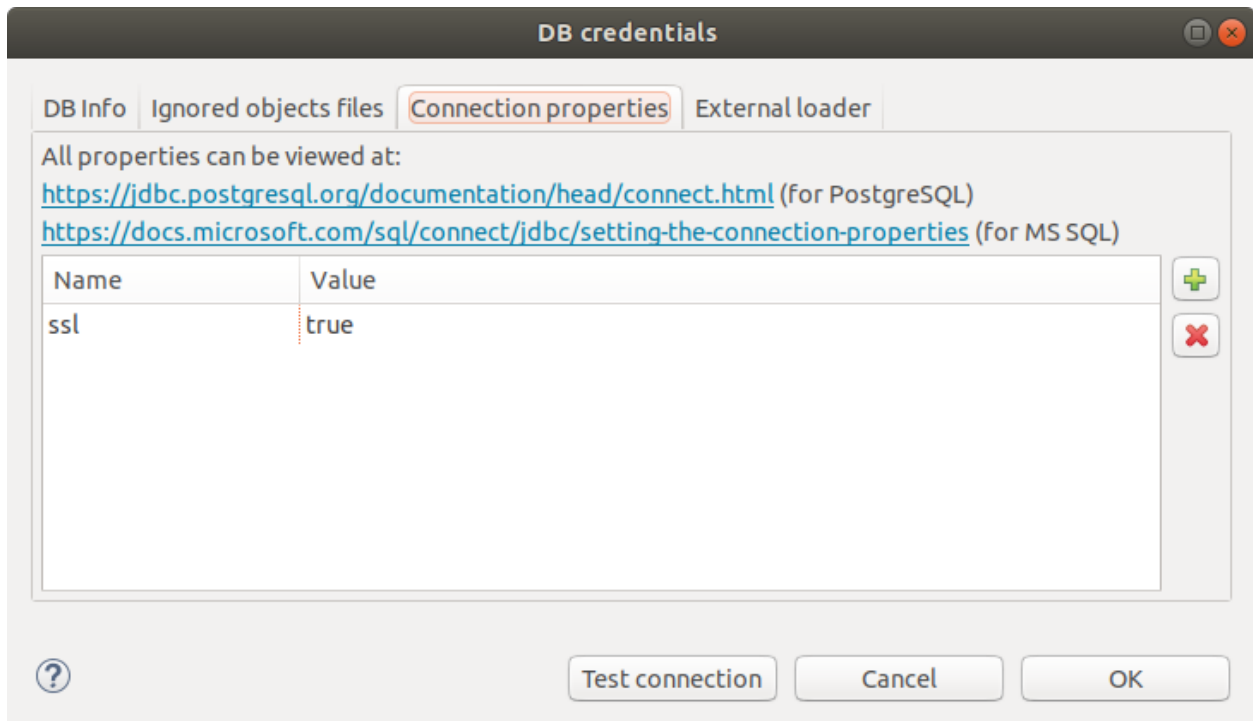
Список исключаемых объектов можно подключить как внешний файл.



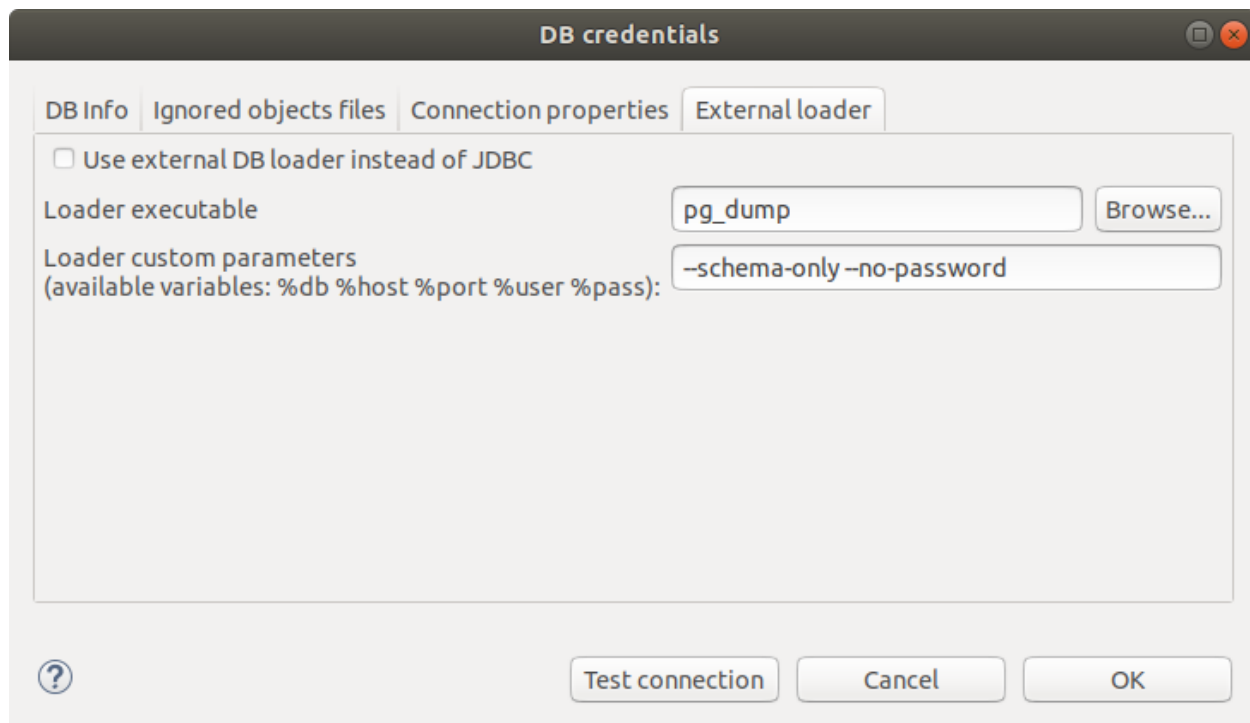
Свойства соединения можно дополнить параметрами указанными по адресу:

<https://jdbc.postgresql.org/documentation/head/connect.html> (Для PostgreSQL)

<https://docs.microsoft.com/ru-ru/sql/connect/jdbc/setting-the-connection-properties> (Для MS SQL)



- **Use external DB loader instead of JDBC / Использовать внешний загрузчик pg_dump** - позволяет использовать внешний загрузчик для базы данных вместо JDBC.
- **Loader executable / Исполняемый файл загрузчика** - путь к утилите, исполняемому файлу или скрипту, используемый для загрузки.
- **Loader custom parameters / Дополнительные параметры загрузчика** - дополнительные параметры вызова утилиты.



Для добавления хранилища на основе данных из уже созданного хранилища следует выделить объект хранилища баз данных, нажать кнопку . Откроется диалог **DB credentials / Данные подключения к БД** в котором можно изменить параметры подключения.

Для добавления хранилища на основе данных из файла **.pgpass**, нажать кнопку . Откроется диалог в котором необходимо выбрать **.pgpass** файл. Далее откроется диалог **Pgpass sources list / Список источников из pgpass**:

Pgpass sources list			
DB Name:	DB User:	DB Host:	Port:
testpgcodekeeper	shamsutdinov_lr	localhost	5432
*	shamsutdinov_lr		5432
*	shamsutdinov_lr		5433
*	shamsutdinov_lr		5434
*	shamsutdinov_lr		5435
*	shamsutdinov_lr		5436
*	shamsutdinov_lr		5437
*	shamsutdinov_lr	10.84.0.6	5432
*	tu_TestAdmin	127.0.0.1	4001
Import Close			

В нем нужно выбрать строку с данными на основе которых будет создаваться хранилище. Откроется диалог **DB credentials** / **Данные подключения к БД** в котором можно подтвердить параметры подключения.

DB credentials	
DB Info	Ignored objects files Connection properties
DB Host:	localhost Port: 5432
DB Name:	testpgcodekeeper
DB User:	shamsutdinov_lr
DB Password:	
Read only:	☐ disallow executing scripts on DB
Entry Name:	shamsutdinov_lr@localhost/testpgcodekeeper ☒ Auto-generate
? Test connection Cancel OK	

После этого можно закрывать диалог **Pgpass sources list** / **Список источников из pgpass**.

Для редактирования хранилища выделить объект хранилища баз данных, нажать кнопку . Откроется диалог **DB credentials** / **Данные подключения к БД** в котором можно изменить параметры подключения.

Для удаления хранилища выбрать необходимую запись и нажать кнопку .

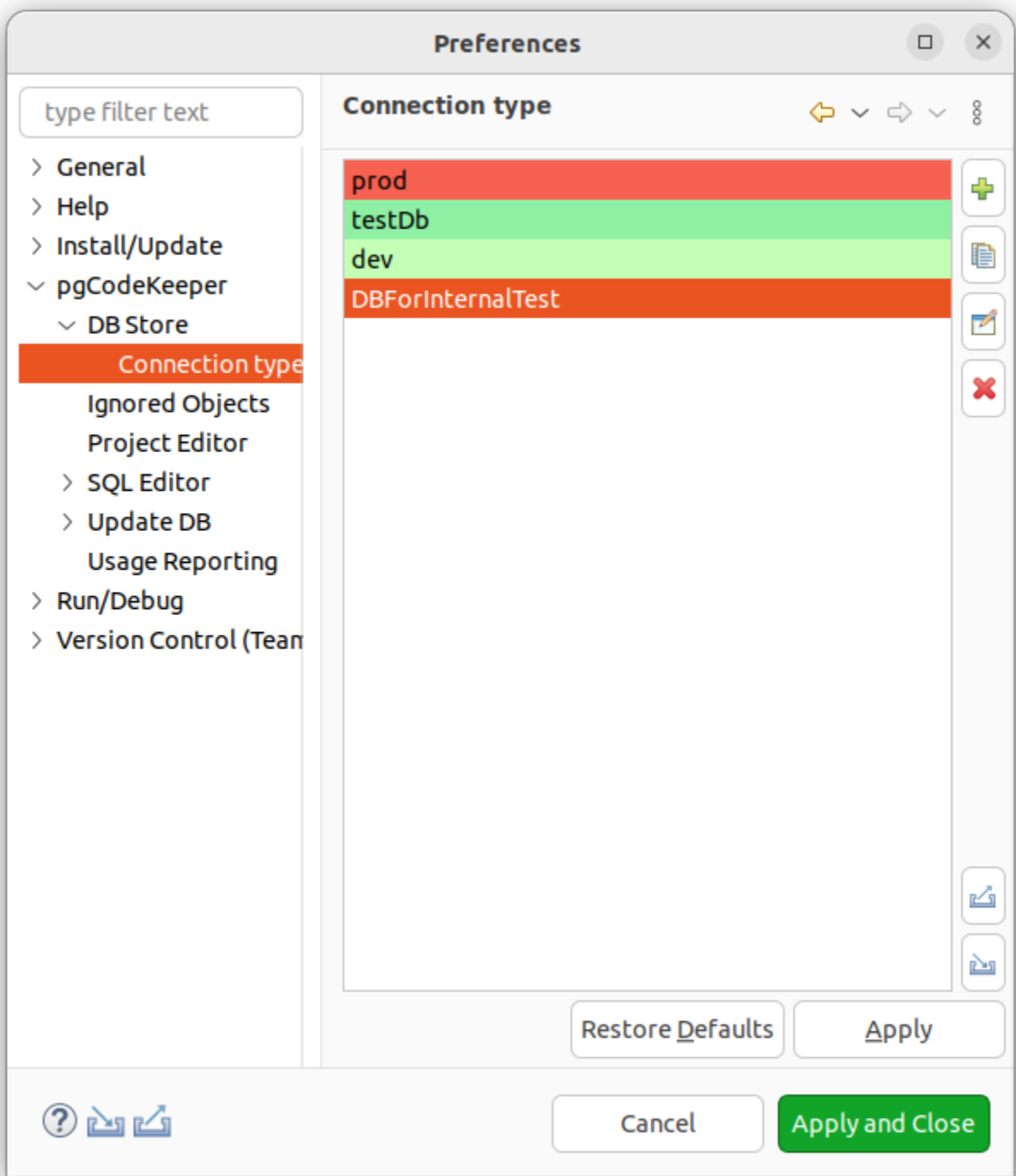
Внимание: Кнопка **Restore Defaults**, на странице настроек, удаляет все записи для подключения к базам данных и создает одну запись default без данных для подключения.

Внимание: Для сохранения внесенных изменений, в хранилище параметров подключения к БД, на странице настроек необходимо нажать кнопку **Apply** или **Apply and Close**.

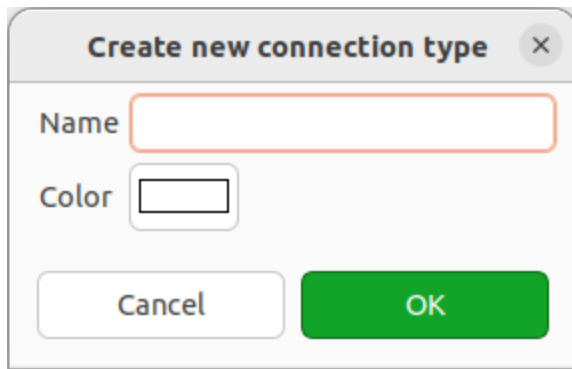
8.11 Тип соединения

Настройки управляющие типом соединения.

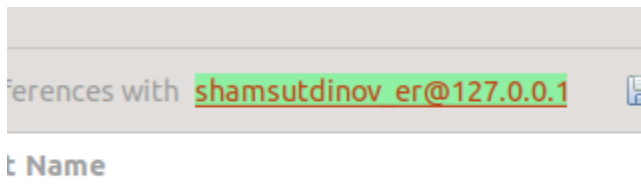
На странице настроек **pgCodeKeeper** -> **DB Store** / **Хранилище БД** -> **Connection type** / **Тип соединения** находится список типов соединения.



Для загрузки списка из файла нажмите по кнопке  **Import** / **Импортировать** и выберите необходимый файл. Также вы можете сохранить список в отдельный файл, для этого нажмите по кнопке  **Export** / **Экспортировать**. Для добавления типа соединения нажмите по кнопке . Откроется диалог **Create new connection type**



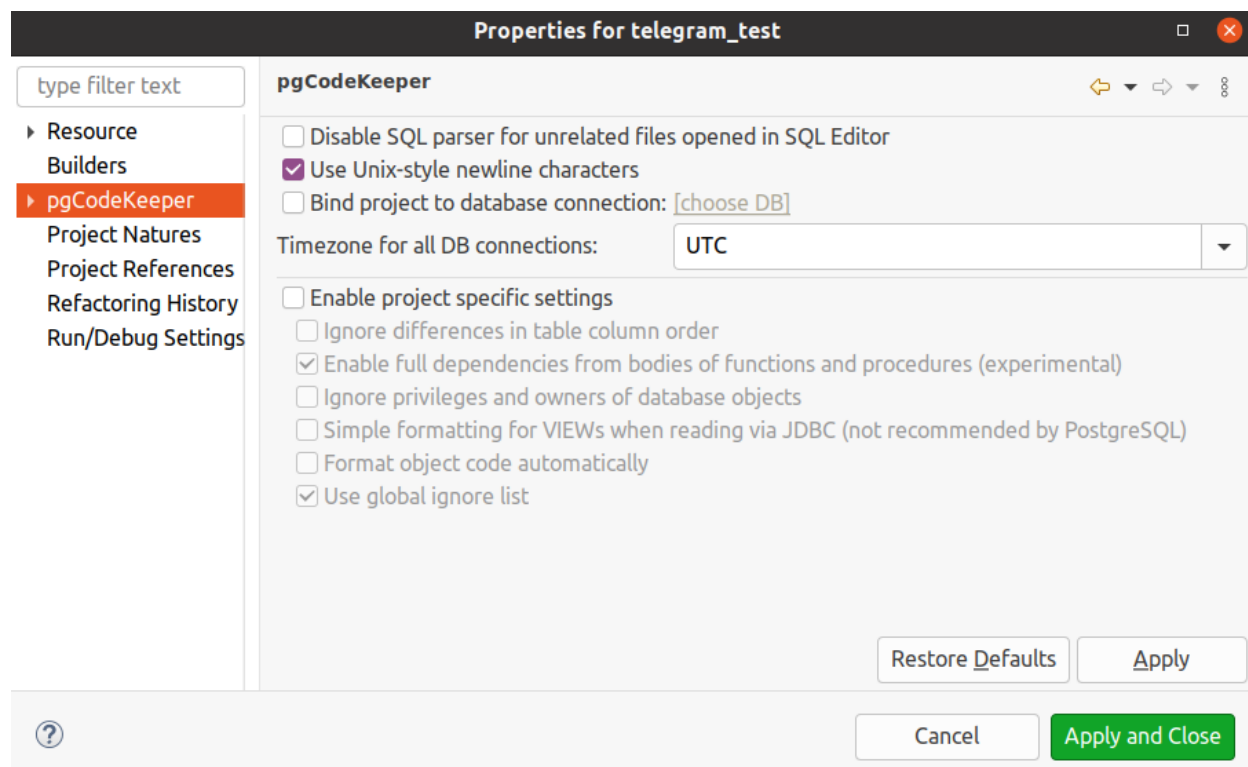
- **Name / Имя** - имя типа соединения, отображаемое в списке и в выпадающем меню диалогового окна **DB credentials DB Store / Хранилище БД**.
- **Color / Цвет** - цвет, используемый для заливки имени соединения в списке *Хранилище БД*, в панели инструментов и SQL редакторе



Настройки проекта

Помимо глобальных настроек, применяемых для всего рабочего пространства, существуют возможность настройки отдельного проекта. Для этого из меню проекта выберите пункт **Properties** или нажмите горячую клавишу (по-умолчанию Alt + Enter). Затем выберите пункт **pgCodeKeeper**.

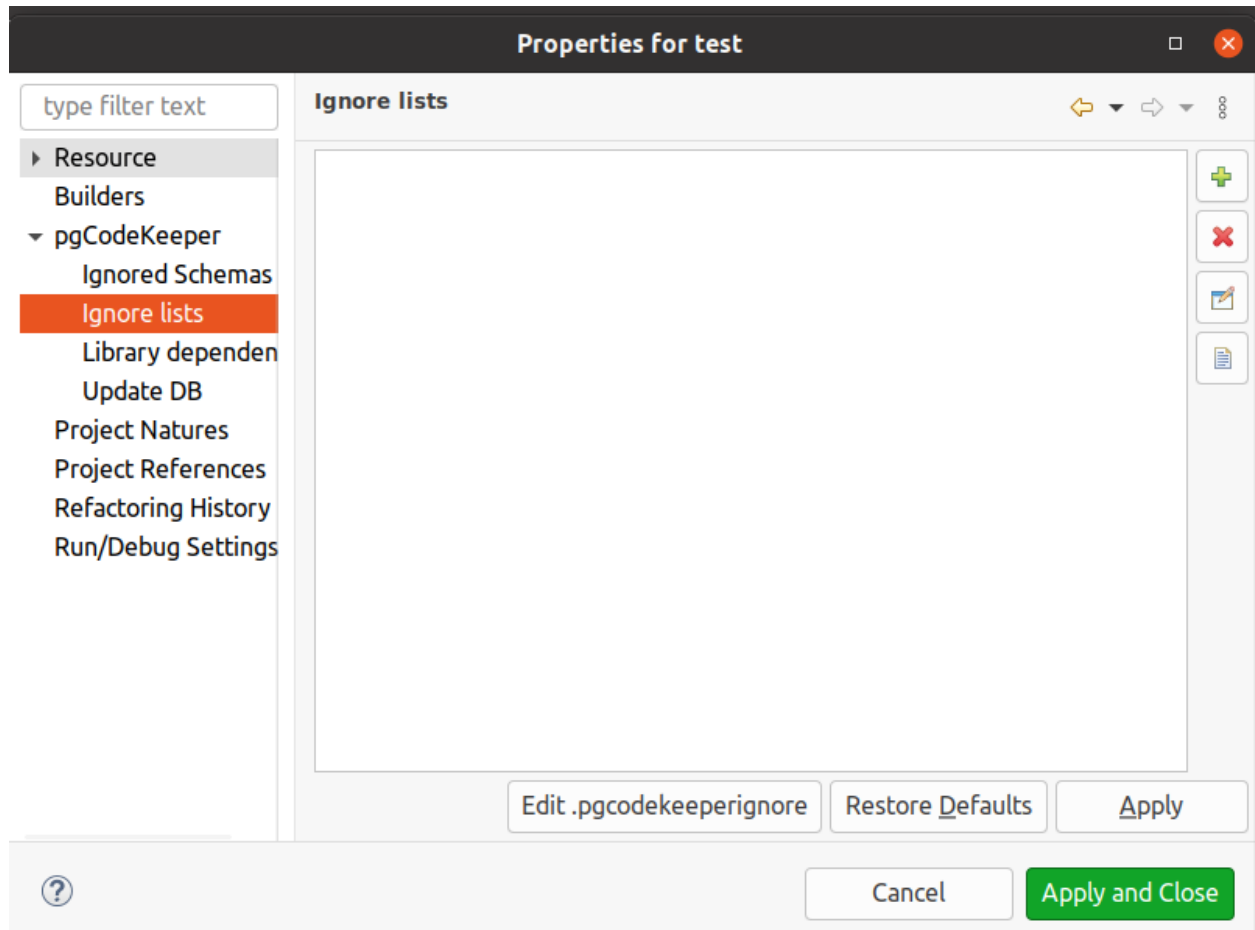
9.1 Основные настройки проекта



- **Disable SQL parser for unrelated files opened in SQL Editor / Отключить SQL парсер для сторонних файлов, открытых в SQL редакторе** - позволяет отключить сборщик pgCodeKeeper для файлов проекта, находящихся вне стандартных директорий. Это отключает поиск ошибок и ссылок на объекты внутри этих файлов.
- **Use Unix-style newline characters / Использовать Unix символы переноса строк** - позволяет использовать Unix символы переноса в строках, например в телах функций, комментариях объектов и т.д.
- **Bind project to database connection / Привязать проект к подключению к БД** - позволяет привязать проект к конкретной БД. Для данного проекта станет невозможно изменить БД на **Панель инструментов Eclipse**.
- **Timezone for all DB connections / Временная зона для всех соединений с БД** - позволяет выбрать, какую временную зону использовать при соединении с базами данных. Отсутствует для MS SQL проектов.
- **Enable project specific settings / Переопределить глобальные настройки (только для текущего проекта)** - позволяет переопределить некоторые глобальные настройки для текущего проекта.
 - **Ignore privileges and owners of database objects / Не учитывать привилегии и владельцев объектов** - позволяет отключить поиск различий в свойствах объектов, связанных с ролями БД.
 - **Ignore differences in table column order / Игнорировать различия в порядке столбцов таблицы** - позволяет не учитывать порядок столбцов при сравнении таблиц.
 - **Enable full dependencies from bodies of functions and procedures (experimental) / Учитывать все зависимости из тел функций и процедур (экспериментально)** - позволяет искать внутри тел функций и процедур зависимости к другим функциям и процедурам.
 - **Simple formatting for VIEWS when reading via JDBC (not recommended by PostgreSQL) / Упрощенное форматирование представлений при чтении через JDBC (не рекомендовано PostgreSQL)** - позволяет использовать упрощенное форматирование представлений, убирая лишние скобки в выражениях. Данный формат может не поддерживаться в будущих версиях PostgreSQL.
 - **Format object code automatically / Форматировать код объектов автоматически** - позволяет отображать и сохранять отформатированный код функций на языках plpgsql и sql согласно настройкам форматирования **SQL Editor/Редактор SQL**. Подробнее про настройки форматирования кода в *Форматирование*.
 - **Use global ignore list / Использовать глобальный список игнорирования** - позволяет включить или отключить использование глобального списка различий со страницы настроек *Исключенные объекты*.

9.2 Списки исключений

Настройки, управляющие списками игнор файлов для данного проекта. Подробное описание работы со списками находится в разделе *Список исключаемых объектов*.

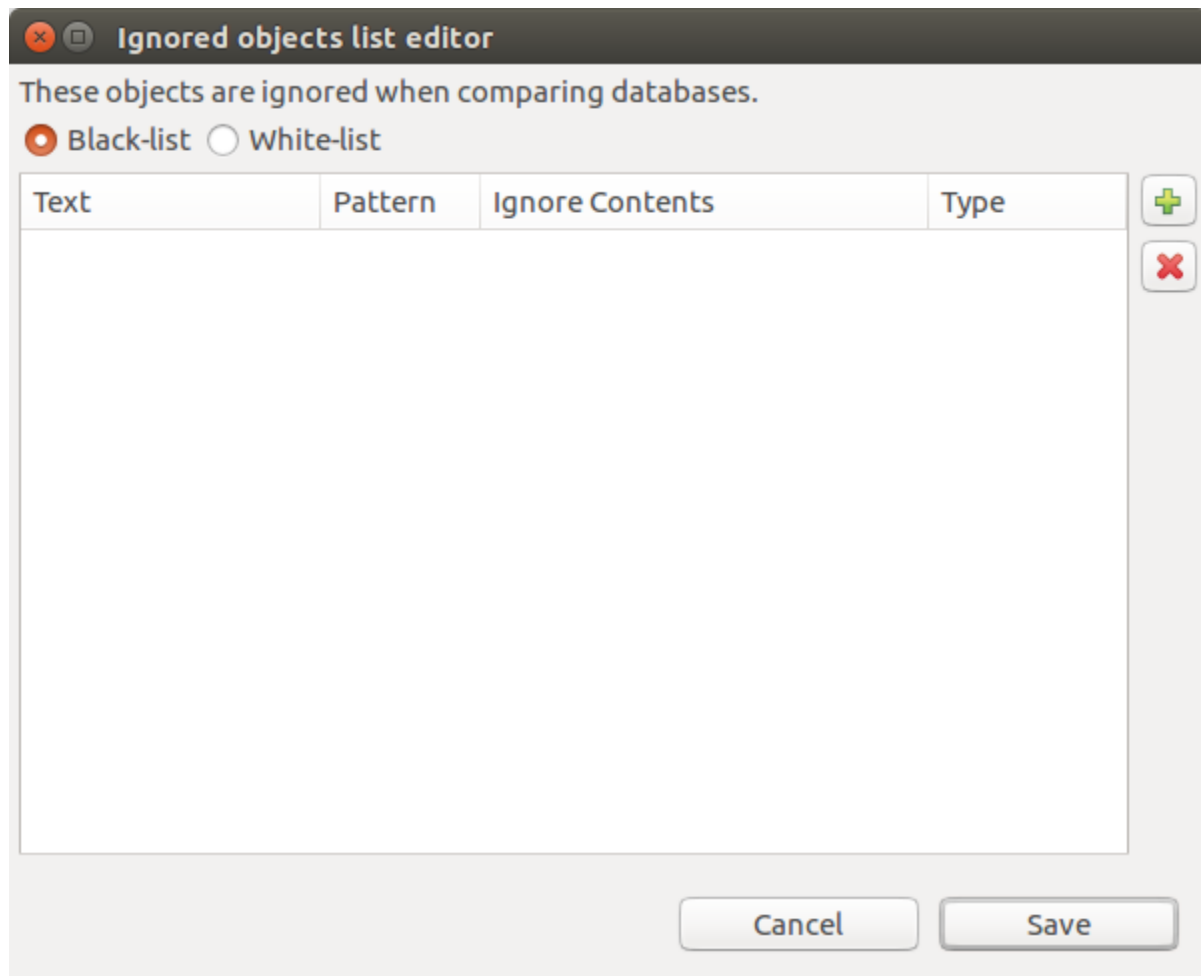


Кнопка **Edit .pgcodekeeperignore** / **Редактировать .pgcodekeeperignore** позволяет изменить стандартный игнор файл, находящийся в корне проекта.

Для добавления существующего файла нажмите кнопку .

Для удаления файла нажмите кнопку .

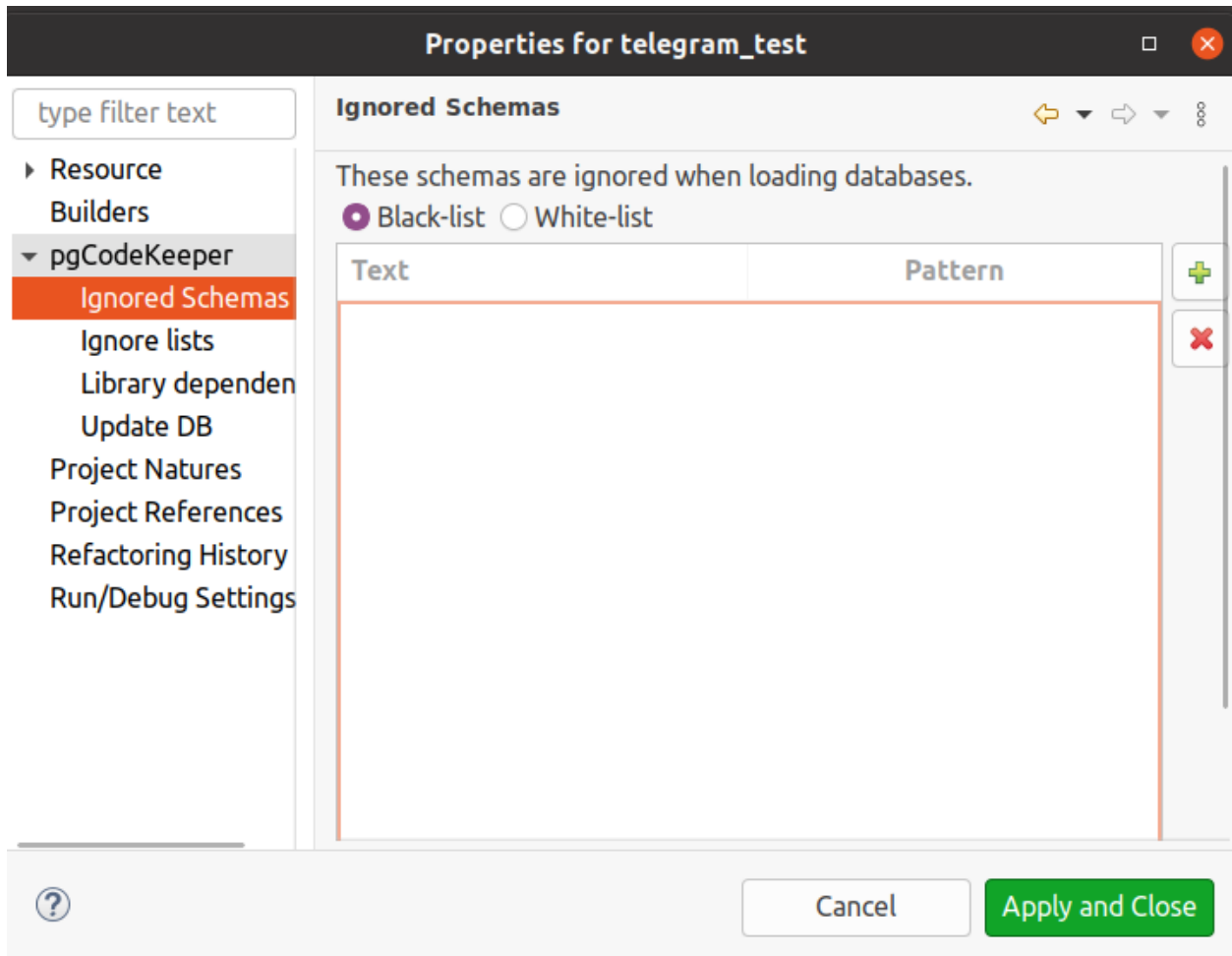
Для создания нового файла нажмите кнопку . Будет открыт редактор списка правил, аналогичный редактору на странице глобальных настроек *Исключенные объекты*.



Для редактирования файла нажмите кнопку . Будет открыт редактор списка правил из текущего файла.

9.3 Исключенные схемы

Настройки, регулирующие загрузку объектов из базы данных. Подробное описание работы находится в разделе *Исключение схем при загрузке*.



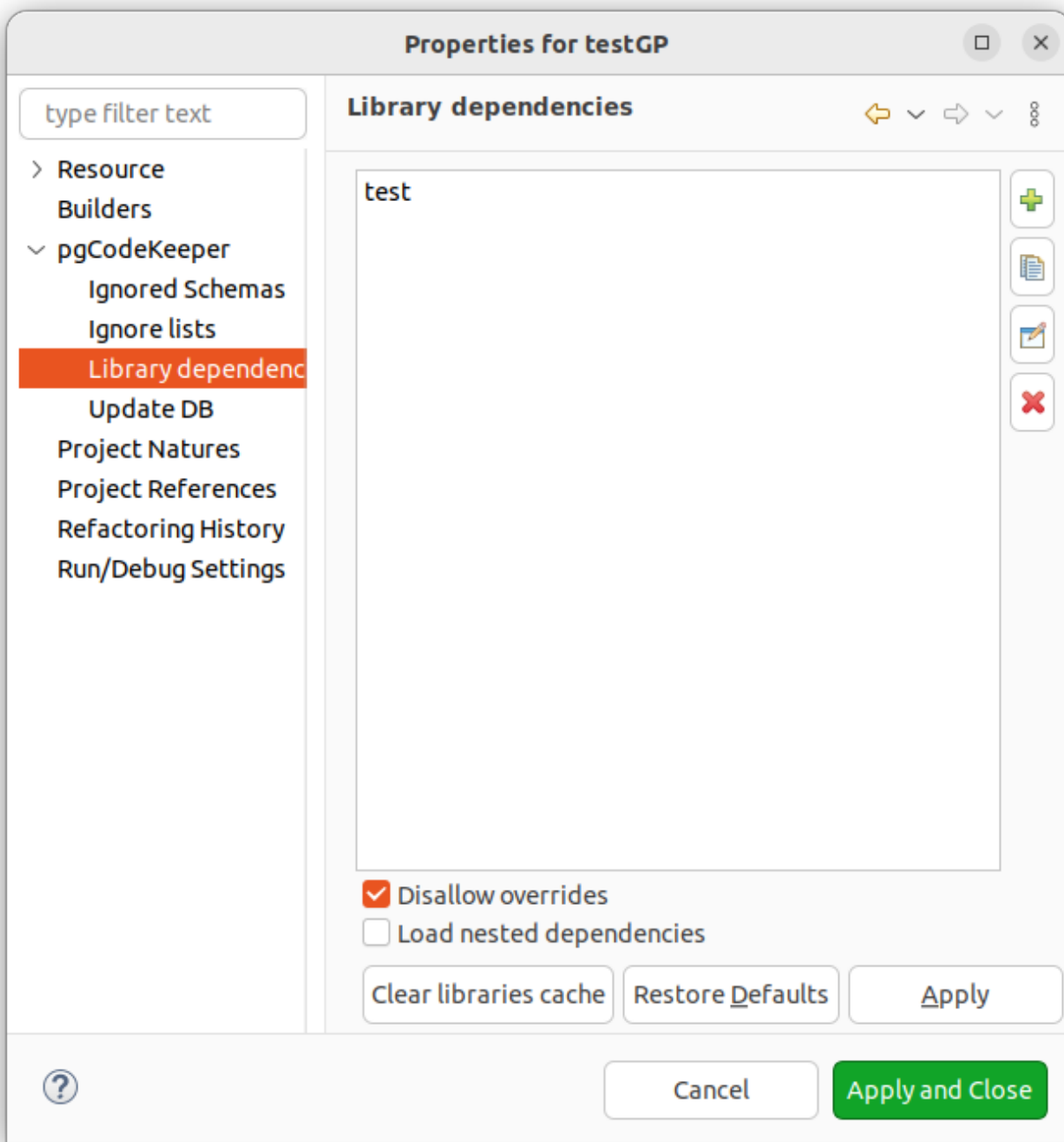
Для добавления объекта нажмите кнопку - откроется редактор добавления нового объекта.

Для удаления объекта нажмите кнопку .

Если возникла необходимость изменить название объекта, надо кликнуть по имени объекта и ввести новое имя.

9.4 Библиотечные зависимости

Настройки, управляющие библиотеками проекта.



Библиотеки - это подключаемые наборы данных объектов, которые «склеиваются» с объектами проекта при сравнении с удаленной базой данных. В качестве библиотек можно использовать директории (это может быть как другой pgCodeKeeper проект, так и обычная директория с файлами), дампы или же удаленная база данных. Кроме того, библиотеки возможно упаковать в zip-архив, загрузить на сервер, и подключить, указав URL архива. Для добавления библиотек воспользуйтесь соответствующими кнопками на боковой панели.

Библиотеки загружаются в порядке, в котором они указаны в списке (самая верхняя библиотека будет загружена первой). Для изменения порядка перетащите элементы при помощи мыши.

Disallow overrides / Запретить переопределение. Не исключена ситуация, когда проект и библиотека (или две библиотеки) содержат объекты с одинаковыми именами. Существует два варианта обработки таких ситуаций. При включенной настройке, такие конфликты вызывают остановку сравне-

ния объектов. Отключенная настройка позволяет игнорировать конфликты: будет использован первый загруженный объект. Независимо от состояния настройки будет отображен вид *Переопределенные объекты*, в котором будут отображены все конфликты.

Load nested dependencies / Загружать вложенные зависимости - настройка, позволяющая загружать библиотеки, на которые ссылаются другие библиотеки (рекурсивная загрузка библиотек). Библиотека с зависимостями должна быть в формате проекта (или архива проекта) и содержать файл `.dependencies` с зависимостями этой библиотеки.

В общем случае, для загрузки зависимостей библиотеки, контейнер этой библиотеки должен иметь включенную настройку `loadNested` в файле `.dependencies`. Например, в цепочке зависимостей `project` → `lib1` → `lib2` → `lib3` для загрузки `lib2` настройка должна быть включена в `project`, а для загрузки `lib3` настройка должна быть включена в `project` и в `lib1`.

Clear libraries cache / Очистить кэш библиотек - позволяет удалить скачанные архивы из кэша.

Диалоговое окно создания/редактирования/копирования библиотечной зависимости выглядит следующим образом.

The image shows a dialog box titled "Create new dependency". It has the following fields and controls:

- Name:** A text input field containing "test".
- Path:** A text input field containing "/home/shamsutdinov_er/runtime-EclipseApplication/testGP". To the right of the field are two buttons: "Relative path" and a file icon button.
- Owner:** A text input field containing "user".
- Ignore privileges/owners:** A checkbox that is checked, with a red border around the entire section.
- Buttons:** At the bottom, there is a help button with a question mark, a "Cancel" button, and a green "OK" button.

- **Name / Имя** - позволяет указать отображаемое имя библиотеки (по умолчанию используется **Path / Путь**).
- **Path / Путь** - позволяет указать источник библиотеки (с поддержкой относительного пути).
- **Owner / Владелец** - позволяет установить нового владельца для всех объектов библиотеки.
- **Ignore privileges/owners / Игнорировать привилегии/владельцев** - позволяет отключить привилегии и владельцев объектов библиотечной зависимости.

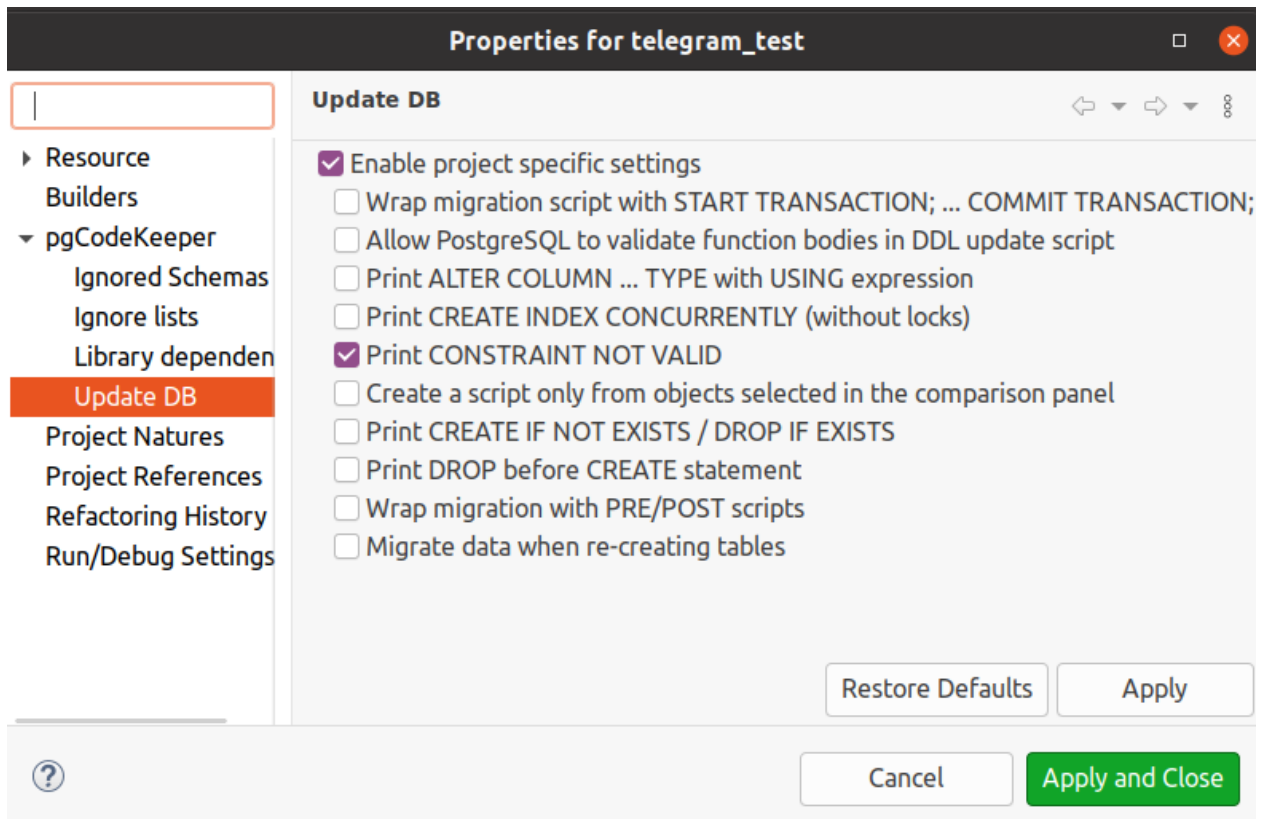
Важно: Если у библиотеки заполнен владелец, он будет отображен вне зависимости от прочих настроек pgCodeKeeper.

Важно: Если у библиотеки не игнорируются привилегии, они будут отображены вне зависимости от

прочих настроек pgCodeKeeper.

9.5 Обновление БД

Настройки, управляющие обновлением базы данных, которые позволяют переопределить некоторые соответствующие настройки со страницы *Обновление БД*.



Обновление проекта

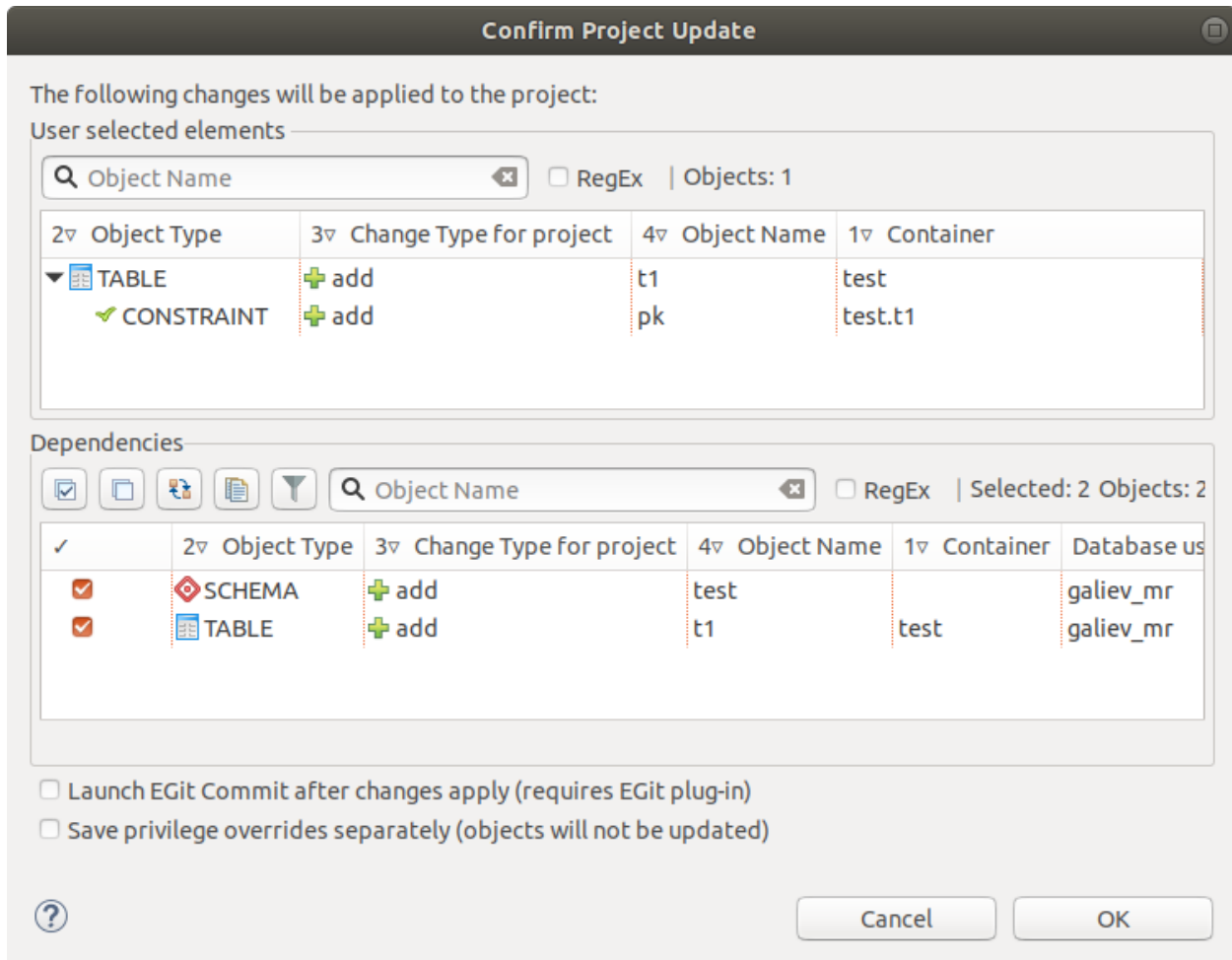
На панели инструментов открываем выпадающее меню кнопки **Get Changes / Получить изменения**  и выбираем источник базы данных, на основании которого будет обновляться проект pgCodeKeeper.

Нажимаем кнопку . После недолгого ожидания на панели различий отобразится список различий с указанием типа изменения.

Object Name				
Differences with 127.0.0.1 into project				
✓	2 Object Type	3 Change type for project	4 Object Name	1 Container
☒	FUNCTION	+ add	sparser_start(numeric, numeric, integer)	public
☐	TABLE	edit	postgres14_table	public
☒	TABLE	edit	postgres14_table3	public
☐	TABLE	delete	foreign_table	public

Открываем выпадающее меню кнопки **Применить** / **Apply**  и выбираем  **Проект** / **Project**.

Отмечаем флажками необходимые изменения, нажимаем на кнопку , и открывается диалог **Подтвердите обновление проекта / Confirm Project Update**.



Проверяем список объектов, выбранных пользователем, и список зависимых объектов. .. important:: Будьте внимательны! Исключение объектов-зависимостей из обновления может привести к ошибкам или неожиданным результатам.

Если вместо изменения самих объектов необходимо сохранить переопределения их свойств (например, привилегии), то выбираем соответствующую опцию. Выполняется только для объектов с типом изменения **edit**.

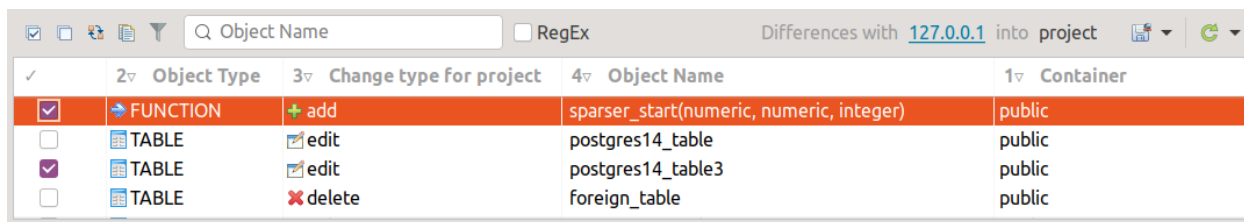
Для подтверждения применения изменений к проекту нажать кнопку **ОК**. В случае успешного применения изменений в **консоли pgCodeKeeper** появится соответствующее сообщение.

Важно: Обновление файлов проекта приводит к сбросу списка различий.

Обновление удаленной базы данных

На панели инструментов редактора проекта открываем выпадающее меню кнопки **Get Changes** / **Получить изменения**  или в выпадающем списке на панели инструментов редактора. Далее выбираем источник базы данных, на основании которого будет сгенерирован DDL скрипт для применения изменений к удаленной базе данных.

Нажимаем кнопку . После недолгого ожидания на панели различий отобразится список объектов с указанием типа изменения.



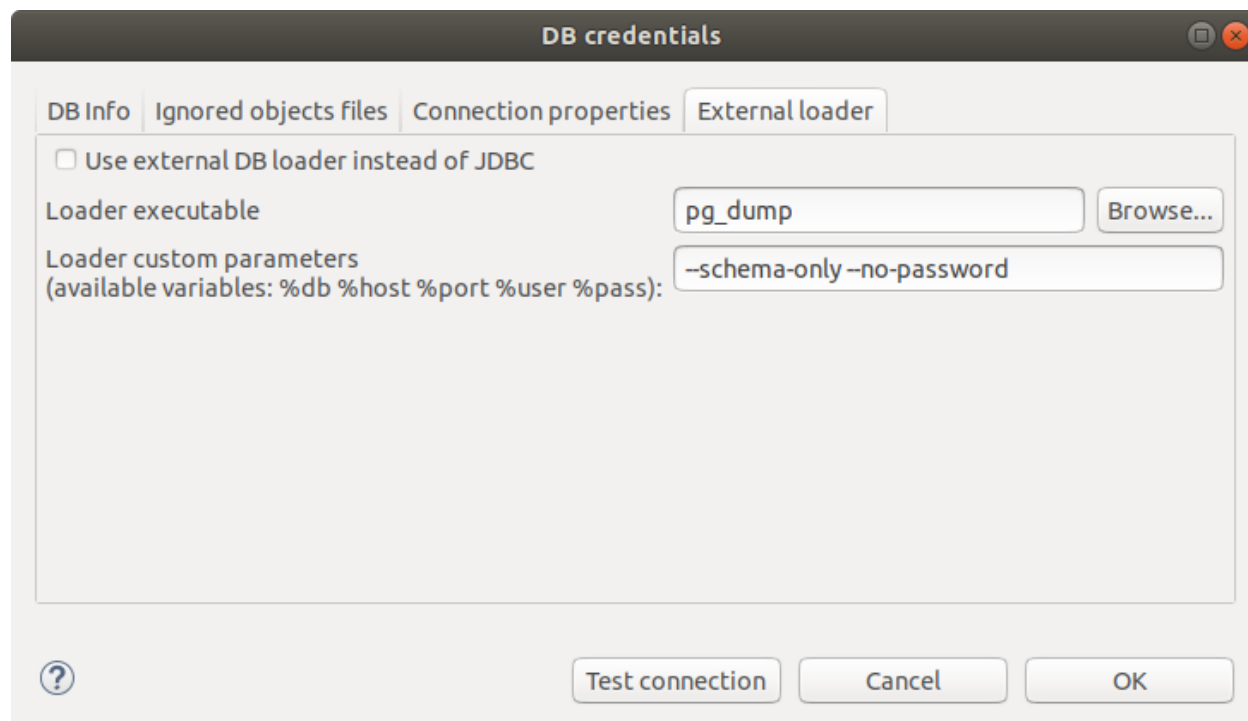
✓	2 Object Type	3 Change type for project	4 Object Name	1 Container
☒	FUNCTION	+ add	sparser_start(numeric, numeric, integer)	public
☐	TABLE	edit	postgres14_table	public
☒	TABLE	edit	postgres14_table3	public
☐	TABLE	delete	foreign_table	public

Открываем выпадающее меню кнопки **Применить** / **Apply**  и выбираем  **БД** / **DB**. Такое же меню выбора направления изменений можно получить кликнув по лейблу project / проект. Отмечаем флажками необходимые изменения, нажимаем на кнопку .

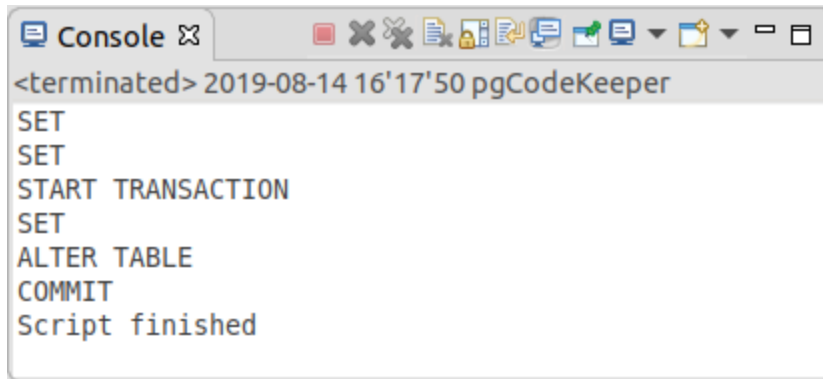
После окончания генерации скрипта откроется SQL редактор. Проверяем сгенерированный скрипт, при необходимости вносим изменения самостоятельно.

```
2018-02-27 09:25:38 migration for default.sql ⓘ
1 SET TIMEZONE TO 'UTC';
2
3 SET check_function_bodies = false;
4
5 START TRANSACTION;
6
7 SET search_path = public, pg_catalog;
8
9 ALTER TABLE t1
10     ADD COLUMN c2 text;
11
12 COMMIT TRANSACTION;
```

Для применения скрипта наката к базе данных необходимо указать способ его выполнения. По умолчанию используется обновление через JDBC. При необходимости использования другой утилиты или параметров запуска можно изменить настройки текущей БД. Подробнее [Хранилище БД](#).



После выбора способа обновления и ввода необходимых параметров нажмите кнопку Обновить DDL или горячей клавишей (по-умолчанию Ctrl+Alt+R). Скрипт наката будет применен для указанной базы данных. Текущий прогресс отображается в консоли pgCodeKeeper.

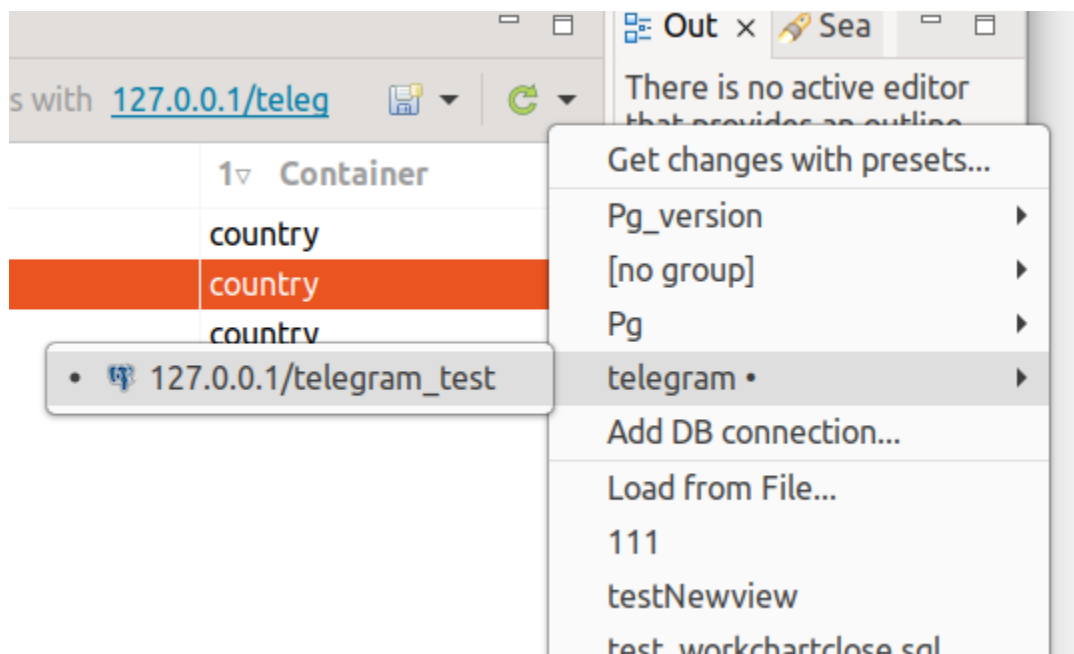


```
<terminated> 2019-08-14 16'17'50 pgCodeKeeper
SET
SET
START TRANSACTION
SET
ALTER TABLE
COMMIT
Script finished
```

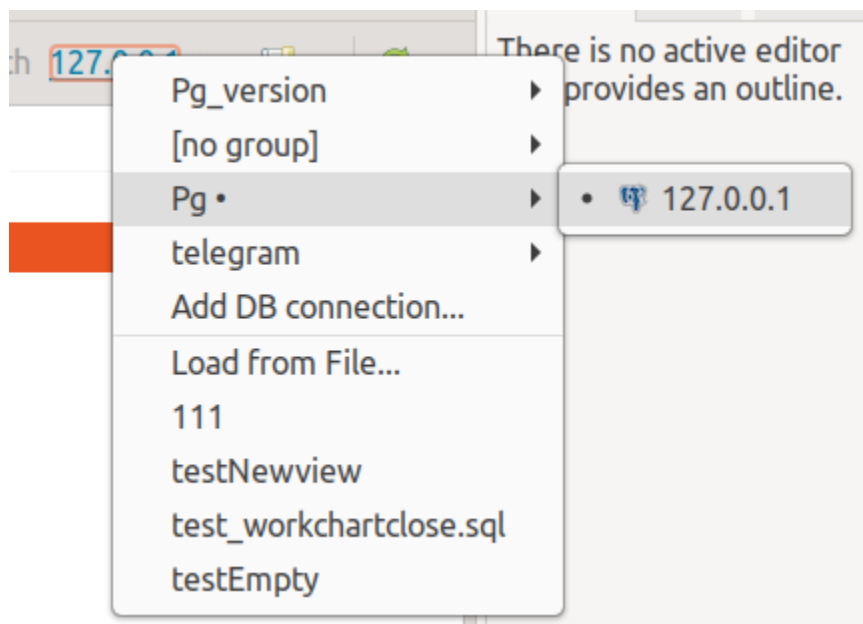
Для большей гибкости имеется возможность наката выделенного фрагмента кода.

Выбор источника базы данных

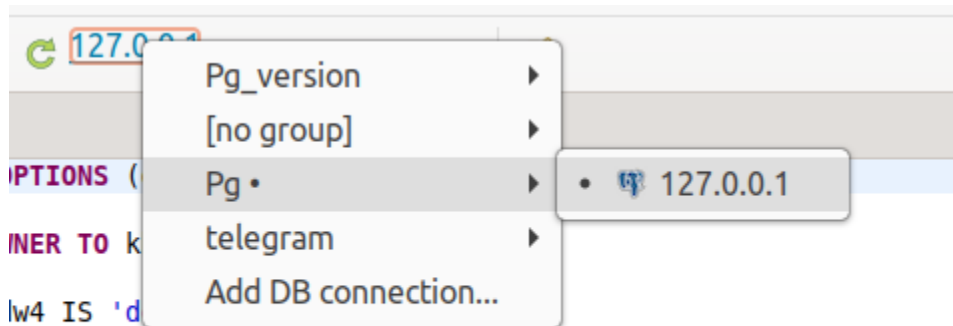
В редакторе проекта для выбора источника БД используется выпадающее меню **Get Changes / Получить изменения** 



или в контекстном меню выбранной БД.



В редакторе SQL выбор источника БД происходит в контекстном меню выбранной БД на панели инструментов Eclipse.



В списках источников БД показаны доступные подключения к БД, объединенные по группам, а также последние использованные дампы-файлы.

Список баз данных находится в *Хранилище БД*, в которое можно перейти по пункту **Add DB connection** / **Добавить подключение к БД** в выпадающем меню кнопки **Get Changes** / **Получить изменения**  или комбобокса.

Для выбора файл-дампа необходимо выбрать элемент списка **Load from file...** / **Загрузить из файла...** В открывшемся диалоговом окне выбираем нужный нам файл и нажимаем кнопку **ОК**. Выбранный файл добавлен в список источников баз данных.

12.1 Группировка источников базы данных

Для объединения баз данных в группы, выберите **DB Store / Хранилище БД** в глобальных настройках pgCodeKeeper и нажмите кнопку **Edit / Редактировать**. В поле **Group / Группа** задайте группу или выберите существующую.

Create new DB connection

DB Info Ignored objects files Connection properties External loader

DB Host: 127.0.0.1 Port: 5432

DB Name:

DB User:

DB Password:

.pgpass passwords are supported

Read only ☐ disallow executing scripts on DB

Database type PG

MS SQL certificate ☒ trust MS SQL certificate

Domain:

Entry Name: 127.0.0.1 ☒ Auto-generate

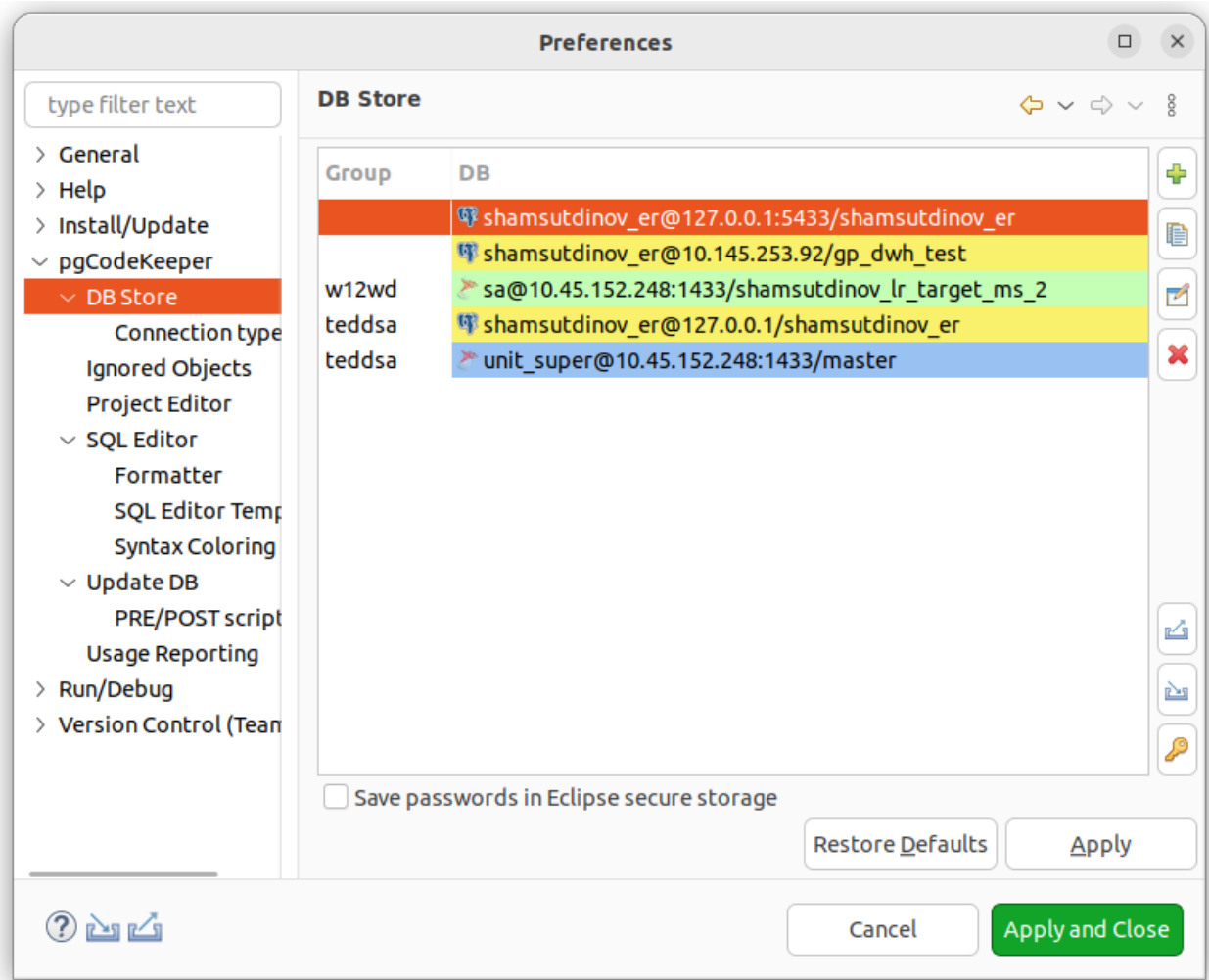
Group:

Connection Type

Test connection Cancel OK

12.1.1 Хранение паролей

При нестабильной работе Eclipse можно отключить хранение паролей в безопасном хранилище Eclipse. В этом случае пароли БД будут храниться открытым текстом в рабочем пространстве Eclipse (workspace). Также, возможно использовать пароли из файла *.pgpass*. Для отключения сохранения паролей в безопасном хранилище Eclipse выключите флажок в настройке **Save password in Eclipse secure storage / Сохранять пароли в безопасном хранилище Eclipse**.



Сортировка списка изменений

Для сортировки объектов в списке изменений необходимо кликнуть по заголовку столбца.

2▼ Change Type	1▼ Object Name
----------------	----------------

По умолчанию используется множественная сортировка. При сортировке по новому столбцу учитывается сортировка по предыдущим выбранным столбцам.

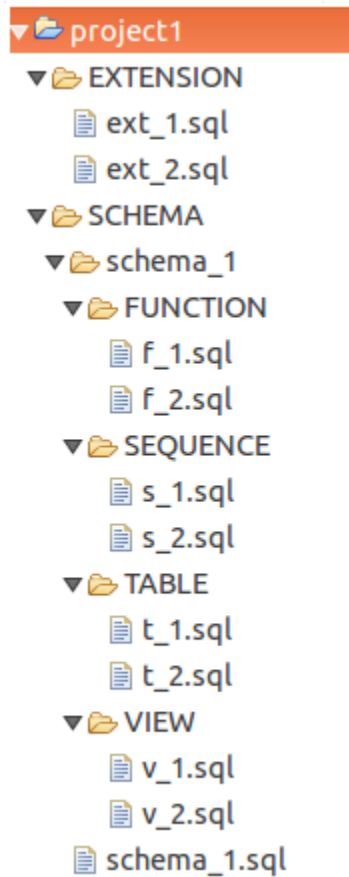
Для сброса сортировки можно кликнуть на заголовок столбца удерживая клавишу Ctrl.

Числа и стрелки в заголовках таблицы показывают порядок и направление сортировки.

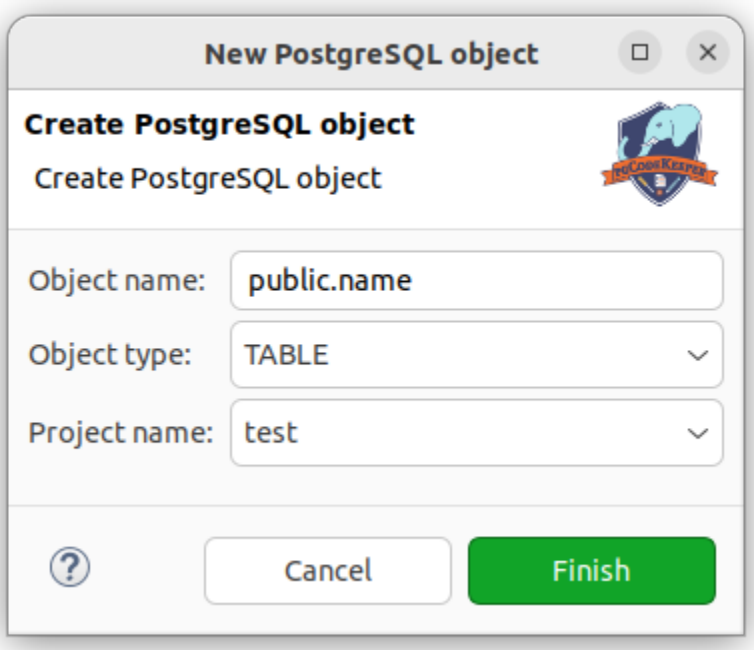
Редактирование pgCodeKeeper проекта

Редактирование проекта состоит в изменении содержимого файлов, а также добавлении и удалении файлов и каталогов находящихся под pgCodeKeeper проектом.

Примечание: Ручное редактирование объектов может привести к ошибкам в работе программы и к некорректному применению изменений к БД.



Для добавления объектов рекомендуем воспользоваться мастером создания SQL объектов: **File -> New -> SQL Object / Объект SQL** или горячей клавишей (по-умолчанию Alt + Insert).



Выберите проект, тип объекта, введите имя и нажмите **Finish**.

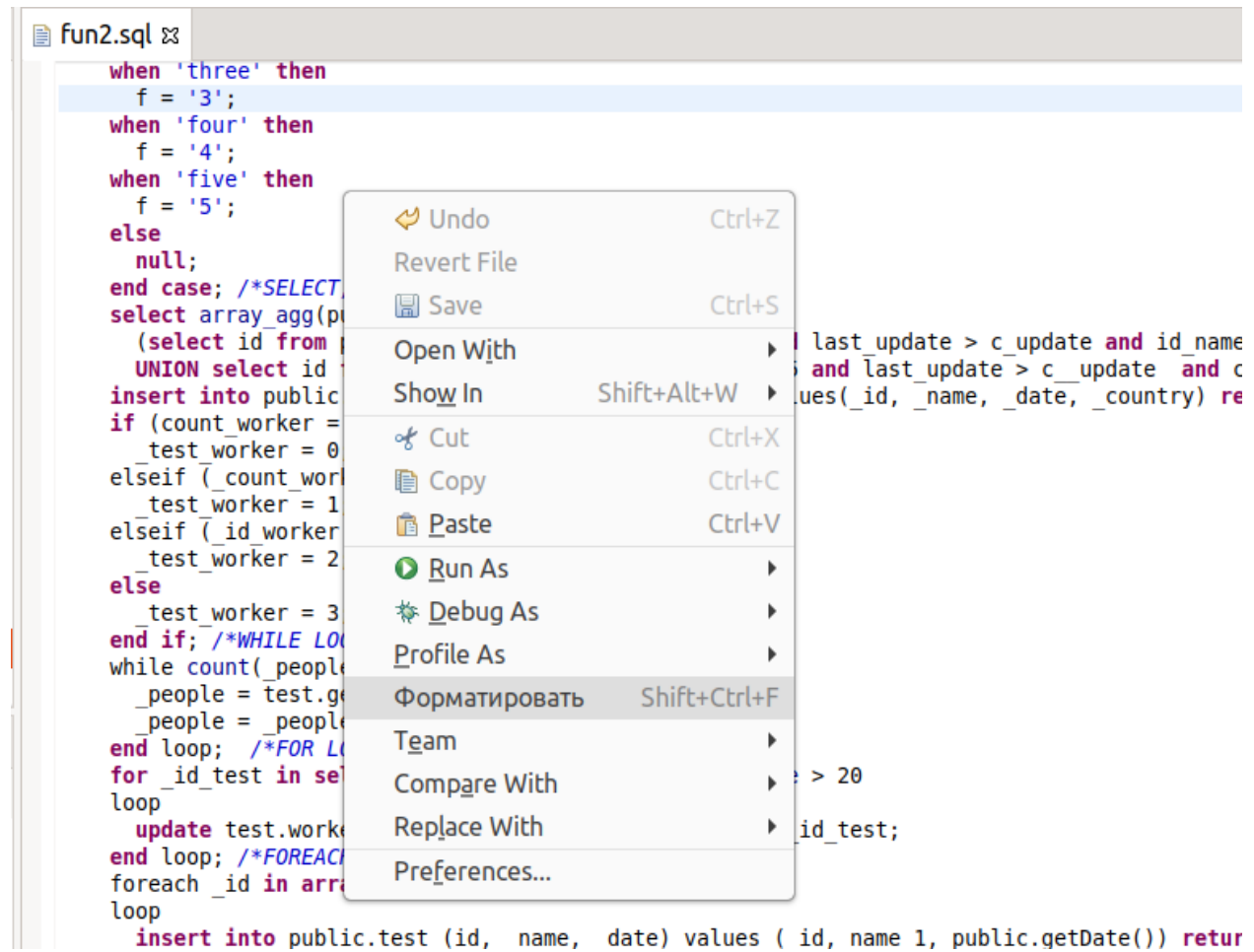
После этого откроется SQL редактор с созданным шаблоном объекта. Если создаваемый объект уже существует, то в редакторе будет открыт этот объект.

Примечание: все возможные данные заполнятся автоматически. Например: если мастер был вызван из контекстного меню файла функции, то автоматически будет выбран текущий проект, тип объекта станет **FUNCTION**, а имя объекта будет частично заполнено именем схемы, в которой находилась функция.

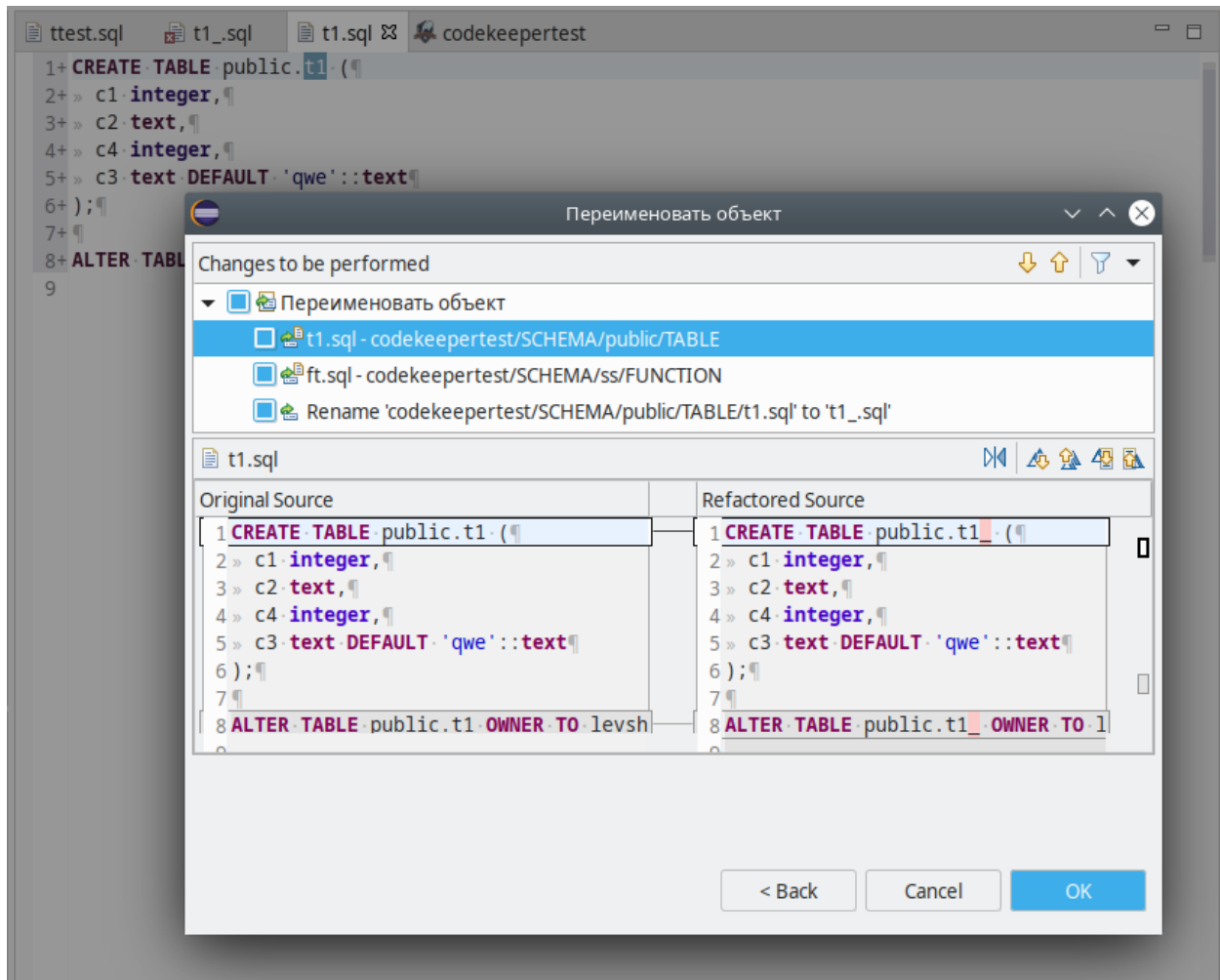
Для редактирования существующего объекта достаточно изменить содержимое его файла в проекте.

Форматирование

Форматирование автоматически применится к коду после использования комбинации клавиш Shift+Ctrl+F, или в контекстном меню выбрать пункт **Format / Форматировать**. Настройки форматирования хранятся в глобальных настройках в *Форматирование*.

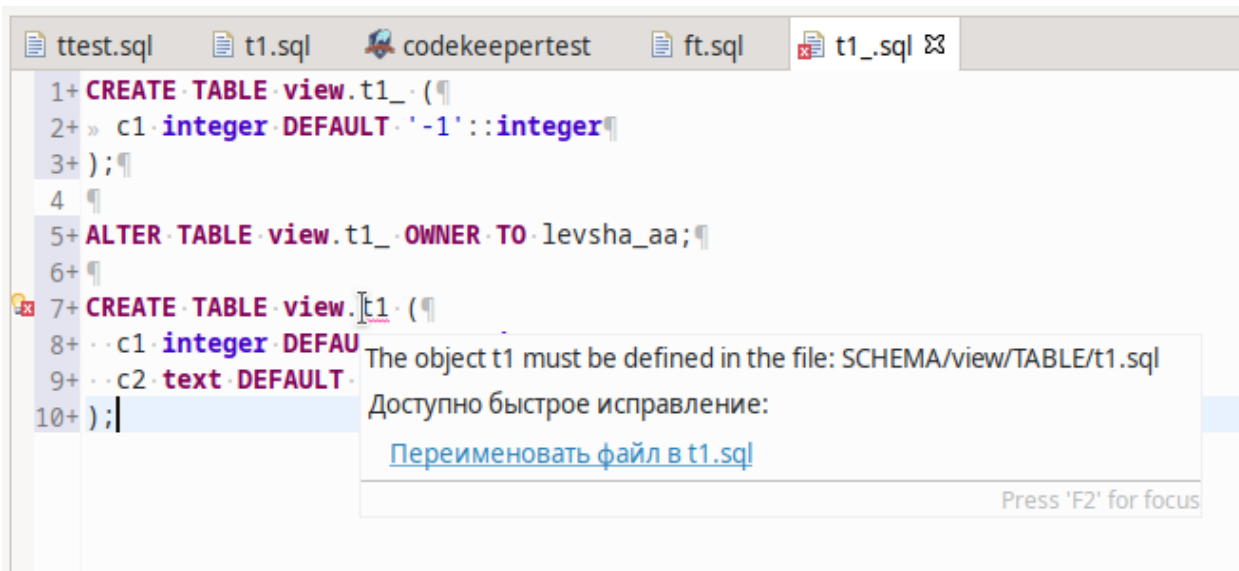


В pgCodeKeeper реализована возможность рефакторинга SQL кода. Объекты в SQL коде можно переименовывать вместе со ссылками на них. Чтобы переименовать объекты выберите в контекстном меню редактора SQL пункт **Переименовать** / **Rename** или нажмите горячую клавишу Alt+Shift+R.



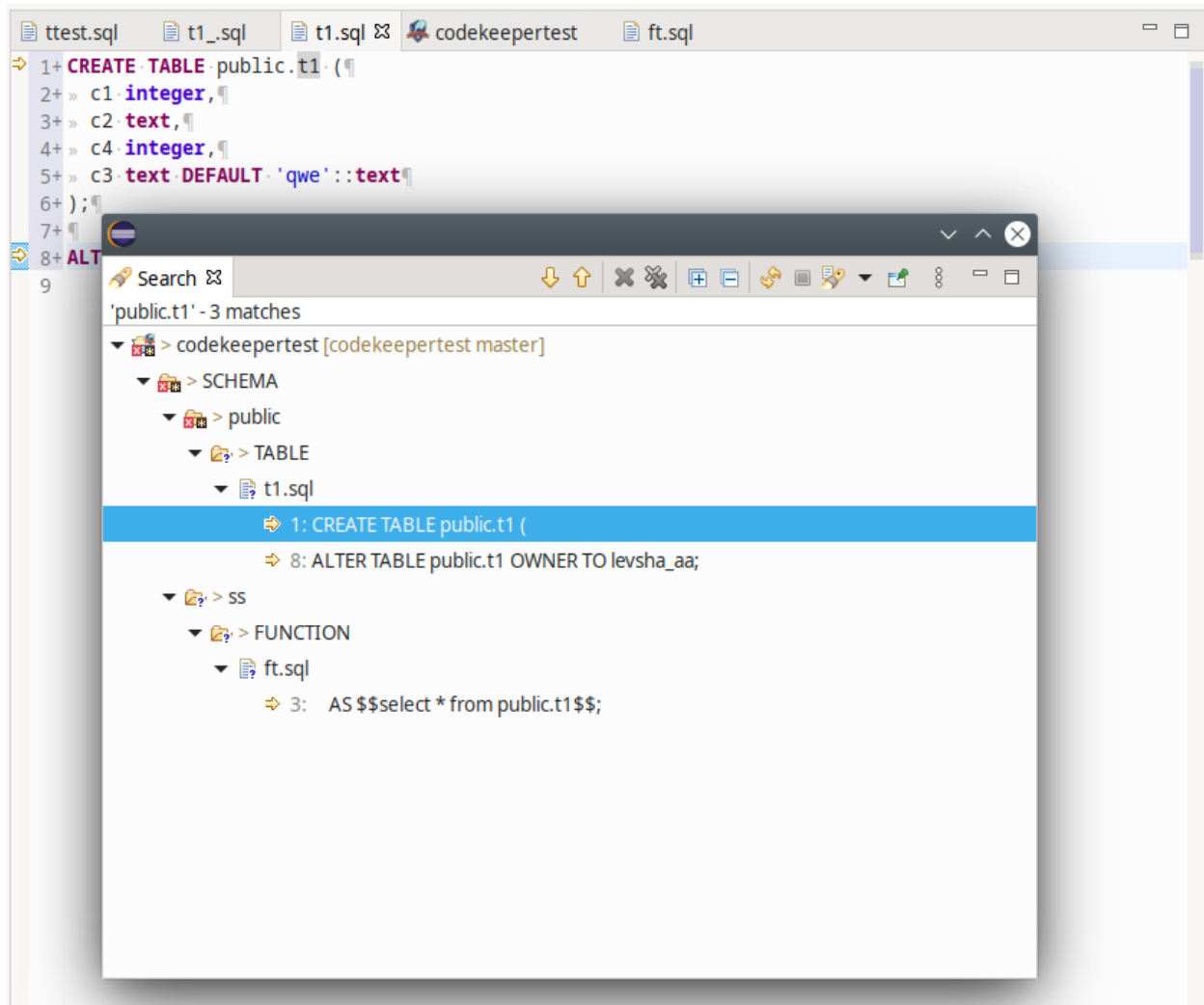
16.1 Быстрые исправления

Также в pgCodeKeepere есть возможность быстрого исправления ошибок некорректного расположения объектов. Для того, чтобы исправить эту ошибку, наведите курсор мыши на ошибку (в имени файла) и во всплывающей подсказке или в контекстном меню Quick Fix редактора SQL выберите пункт **Переименовать файл / Rename file**. Файл объекта будет перемещен на ожидаемое для него место в проекте.



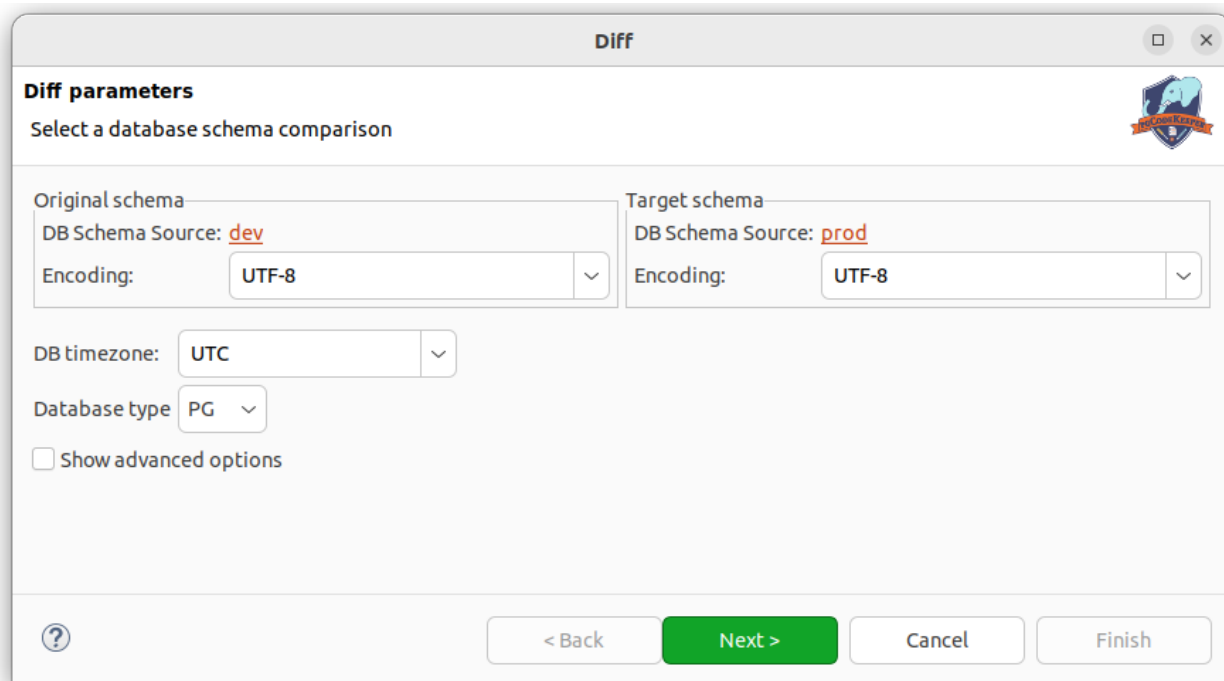
17.1 Поиск ссылок на объекты

Для поиска всех ссылок на необходимый объект, необходимо выделить этот объект и воспользоваться горячей клавишей **Ctrl+Shift+G** или в контекстном меню редактора SQL выбрать **Find references / Найти ссылки**



Сравнение схем БД

- Открываем мастер сравнения баз данных: **pgCodeKeeper - Diff Wizard** / Мастер сравнения или вызываем контекстное меню pgCodeKeeper проекта и выбираем **pgCodeKeeper - Diff...** / Сравнить....



- Выбираем источник исходной схемы базы данных, которую нужно обновить, и целевой схемы базы данных, на основании которой обновляем исходную базу данных.
- Выбираем кодировки источников и временную зону БД.
- Выбираем тип БД **Database type** / Тип базы данных.

- При необходимости переопределить глобальные настройки можно воспользоваться опцией **Show advanced options** / **Показать дополнительные опции**.
- Нажимаем кнопку **Next**.

Diff

Diff table

Select the items for updating.



Original schema:

Target schema:

☒ ☐ ☐ ☐ ☐

☐ RegEx | Selected: 0 Objects: 3

✓	2▼ Object Type	4▼ Change Type	1▼ Object Name	3▼ Container
☐	→ FUNCTION	edit	"incre'ment"(integer)	public
☐	→ FUNCTION	edit	serv()	public
☐	→ VIEW	delete	vx1	public

?

< Back

Next >

Cancel

Finish

В таблице сравнения отмечаем изменения, которые необходимо применить. Примечание: панель инструментов работает аналогично панели инструментов редактора проектов pgCodeKeeper.

При нажатии кнопки **Finish** откроется SQL редактор со скриптом миграции, сгенерированным для выбранных изменений.

Генерация тестовых данных

Существует возможность генерации тестовых данных для таблицы в виде **Insert** конструкции:

Для этого откройте мастер создания нового объекта **File -> New -> Others** и выберите категорию **pgCodeKeeper -> Сформировать тестовые данные** и нажмите **Next**.

Create data batch

Create data batch for a table

Table name:

Row count:

Name	Type	Generator
c1	bigint	RANDOM
c2	bit(1)	RANDOM
c3	boolean	RANDOM
c4	character(1)	RANDOM
c5	date	RANDOM
c6	timestamp without time zone	RANDOM
c7	double precision	RANDOM
c8	integer	RANDOM
c9	json	RANDOM
c10	numeric	RANDOM
c11	real	RANDOM
c12	smallint	RANDOM
c13	text	RANDOM
c14	time without time zone	RANDOM
c15	numeric(3,0)	RANDOM
c16	character varying(20)	RANDOM
c17	character varying(4)	RANDOM
c18	interval	ANY
c19	timestamp with time zone	RANDOM

Column Name:

Type:

Generator:

☐ UNIQUE ☒ NOT NULL

Range start:

Range end:

☒ Explicit type cast

Для каждой колонки в зависимости от типа и генератора доступны следующие настройки:

- **Имя колонки** - имя текущей колонки.
- **Тип колонки** - тип данных колонки. Для неподдерживаемых типов существует тип OTHER.
- **Генератор** - алгоритм генерации данных. На данный момент их 4:

RANDOM

Случайное значение из диапазона.

INCREMENT

Последовательное увеличение стартового значения на значение инкремента.

CONSTANT

Константное значение. Текстовые значения будут заключены в одинарные кавычки.

ANY

Любое значение (включая вызовы функций). Значение не будет отформатировано.

- **Начало диапазона** - стартовое значение

- **Конец диапазона** - конечное значение
- **Шаг инкремента** - значение шага инкремента
- **Произвольное выражение** - произвольное значение, которое будет записано без форматирования.

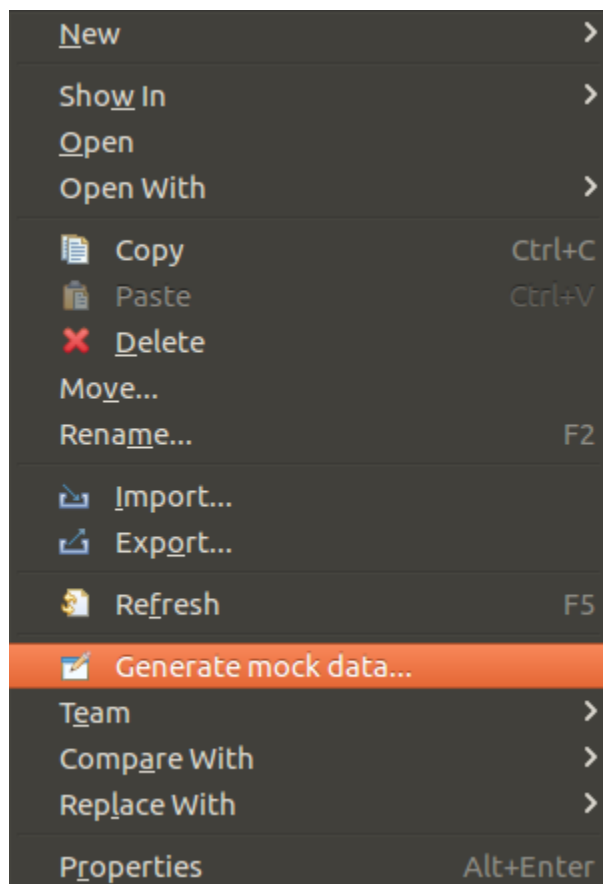
Для управления списком колонок доступны следующие кнопки:

- **Добавить колонку** добавляет колонку типа INTEGER с генератором RANDOM.
- **Удалить колонку** удаляет выбранные колонки.
- **Вверх** передвигает выбранную колонку вверх в списке.
- **Вниз** передвигает выбранную колонку вниз в списке.
- **Удалить необязательные** удаляет колонки у которых нет свойства NOT NULL.

Настройка **Явное приведение типов** добавляет явное приведение типов для каждого значения (например, 7::integer).

После нажатия **Finish** будет открыт SQL редактор с готовым Insert запросом.

Примечание: Также имеется возможность вызвать данный мастер из контекстного меню файла таблицы. В этом случае данные заполнятся автоматически.



Список исключаемых объектов

Список исключаемых объектов - это список объектов, которые не должны или наоборот должны отображаться при сравнении баз данных.

Внимание: Объекты, отфильтрованные данными списками, участвуют в сравнении и могут попасть в итоговый скрипт миграции по зависимости.

Существует два типа списков исключаемых объектов:

черный список

разрешает отображение всех объектов, кроме указанных в правилах списка;

белый список

работает противоположно черному, запрещает отображение всех объектов, кроме указанных в правилах списка.

В каждом проекте можно создать **список исключаемых объектов**, распространяющий свое действие только на данный проект, для этого в корне проекта необходимо создать файл **.pgcodekeeperignore** и заполнить его правилами согласно синтаксису списков.

Внимание: Формат файла **.pgcodekeeperignore** чувствителен к регистру. Кодировка должна быть – UTF-8.

Ниже, в разделе *Синтаксис списков* рассмотрим правила формирования файла **.pgcodekeeperignore**.

Примечание: В дополнении к файлу **.pgcodekeeperignore**, в графической версии pgCodeKeeper существует **общий список исключаемых объектов**. Этот список распространяет свое действие на все проекты, находящиеся в директории проектов. Заполнить его можно на странице с глобальными настройками в разделе *Исключенные объекты*.

20.1 Синтаксис списков

Файл `.pgcodekeeperignore` состоит из двух частей:

заголовок

является обязательным правилом, занимает первую строку и определяет тип списка;

правила

строки правил запрещающие или разрешающие (в зависимости от типа списка) показывать какой-либо объект.

Черный список позволяет отображать только те объекты, которые не указаны в правилах. Он построен таким образом, что заголовок списка разрешает показывать все объекты, а последующие за ним правила исключают объекты из отображения.

```
# комментарий
SHOW ALL
HIDE flag [, ...] objectName [ db=dbName ] [ type=objectType [, ...] ] # комментарий
[ ... ]
```

Описание частей черного списка

Строка файла <code>.pgcodekeeperignore</code>	Описание
SHOW ALL	[заголовок] разрешает показывать все объекты
HIDE flag [, ...] objectName [db=dbName] [type=objectType [, ...]]	[правило] исключает из отображения какой-либо объект

Белый список работает от обратного, позволяя отображать только те объекты, которые указаны в правилах. Он построен таким образом, что заголовок списка запрещает показывать все объекты, а последующие за ним правила разрешают объекты для отображения.

```
# комментарий
HIDE ALL
SHOW flag [, ...] objectName [ db=dbName ] [ type=objectType [, ...] ] # комментарий
[ ... ]
```

Описание частей белого списка

Строка файла .pgcodekeeperignore	Описание
HIDE ALL	[заголовок] запрещает показывать все объекты
SHOW flag [, ...] objectName [db=dbName] [type=objectType [, ...]]	[правило] разрешает для отображения какой-либо объект

Где:

objectName, **dbName**, **objectType**

идентификаторы для поиска объекта, аналогичные идентификаторам PostgreSQL для использования специальных символов (например, в регулярных выражениях) идентификатор должен быть заключен в кавычки
подробнее см. правило **identifier** ниже

objectName	название искомого объекта (является обязательной частью правила)
dbName	имя базы данных в которой будет искаться объект (опционально)
objectType	тип искомого объекта (опционально)

Важно: **objectType** может быть равен одному из следующих значений: CAST, USER, ROLE, ASSEMBLY, SCHEMA, EXTENSION, TYPE, DOMAIN, SEQUENCE, OPERATOR, FTS_PARSER, FTS_TEMPLATE, FTS_DICTIONARY, FTS_CONFIGURATION, TABLE, FUNCTION, PROCEDURE, AGGREGATE, CONSTRAINT, VIEW, INDEX, TRIGGER, RULE, POLICY.

identifier

строка-идентификатор без кавычек, состоящая из латинских букв, цифр и подчеркиваний;
не может начинаться с цифры;
строка идентификатор не может быть ключевым словом (зарезервированной директивой языка)
или
строка, ограниченная двойными кавычками " или апострофами ' (ограничители в начале и в конце строки должны совпадать);
ограничители, содержащиеся в строке экранируются повтором символа (например: "1""2'3"
или '1"2' '3')

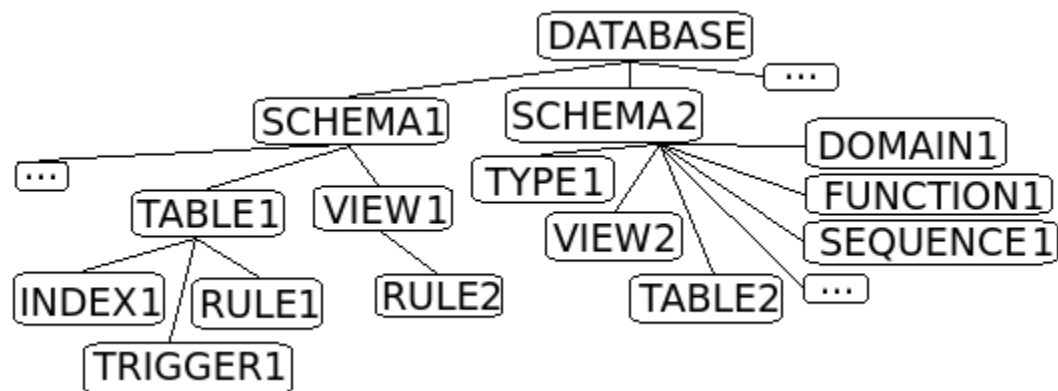
Внимание: «objectType» чувствителен к регистру.

flag { NONE | REGEX | CONTENT | QUALIFIED }

флаг способа поиска объекта

REGEX	воспринимает objectName как регулярное выражение ¹
CONTENT	применяет правило для совпавшего объекта и всего его содержимого
QUALIFIED	сравнивает objectName с квалифицированным именем объекта
NONE	строгое совпадение без регулярного выражения и без содержимого

Содержимое - это содержимое объекта согласно иерархии проекта pgCodeKeeper:



Например для «TABLE1» содержимым будет являться «INDEX1», «TRIGGER1» и «RULE1».

Дополнительные опции:

- db=dbName правило будет применено только к БД, имя которой подходит под регулярное выражение, заданное этим параметром
- type=objectType правило будет применено только к объектам с указанным типом

Примечание: Для указания нескольких типов для одного и того же объекта необходимо написать их через запятую.

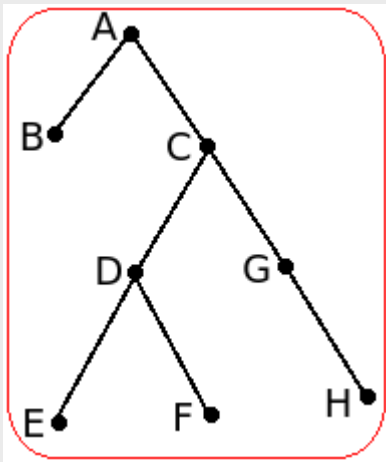
Внимание: Название базы данных чувствительно к регистру.

Примеры для белого списка

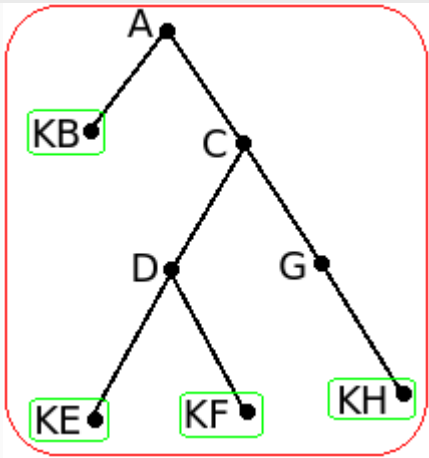
¹ Флаг **REGEX** позволяет правилу производить поиск частичных совпадений с регулярным выражением.

Пример и описание	Область действия
-------------------	------------------

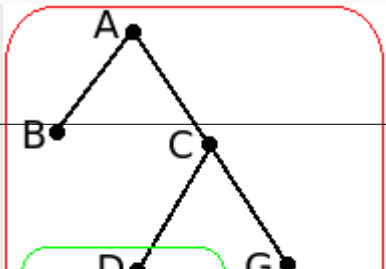
HIDE ALL
[заголовок]
запрещает отображение
всех объектов



HIDE ALL
[заголовок]
запрещает отображение
всех объектов



SHOW REGEX K
[правило]
разрешает отображать
объекты которые частично
попадают под регулярное
выражение «K»

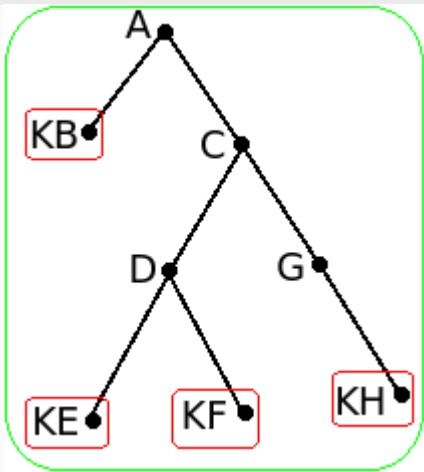
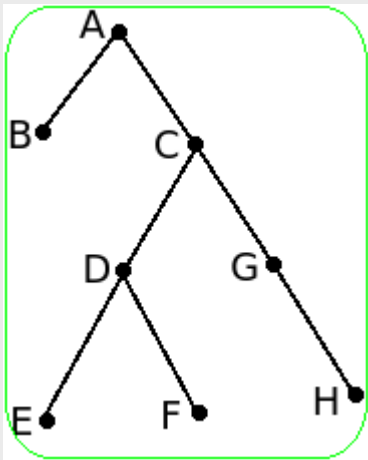


Примеры для черного списка

Пример и описание	Область действия
-------------------	------------------

SHOW ALL

[заголовок]
разрешает показывать
все объекты

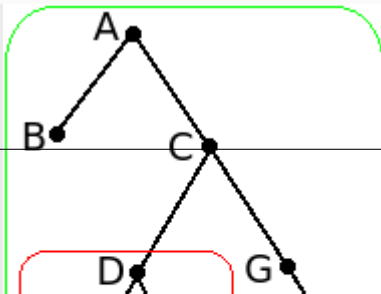


SHOW ALL

[заголовок]
разрешает показывать
все объекты

HIDE REGEX K

[правило]
исключает объекты которые
частично попадают под
регулярное выражение «K»



Ключевые слова:

HIDE SHOW ALL REGEX CONTENT QUALIFIED NONE

Эти слова не могут быть идентификаторами, для их использования они должны быть взяты в кавычки. Зарезервированы только слова, полностью совпадающие по регистру, например Content – разрешенный идентификатор.

Пример исключения объекта название, которого полностью совпадает (в том числе и по регистру) с ключевым словом **SHOW**:

```
SHOW ALL
HIDE NONE "SHOW"
```

20.2 Совместное использование черного и белого списков

Черные и белые списки могут использоваться вместе. В таком случае, их правила объединяются в один общий список. Правила, контролирующие отображение одного и того же объекта, складываются в одно общее правило по следующим принципам:

- если «широта» правил различается, то преобладает более широкое правило (включающее в себя сам объект и его содержимое)
- если «широта» правил одинакова, то преобладает скрывающее объект правило

«Широта» правила - это включение или не включение, в область действия правила, содержимого того или иного объекта, т.е. состояние флага CONTENT (для **общего списка** графической версии pgCodeKeeper, это опция «Игнорировать содержимое» описанная в разделе *Исключенные объекты*).

Пример совместного использования черного и белого списков:

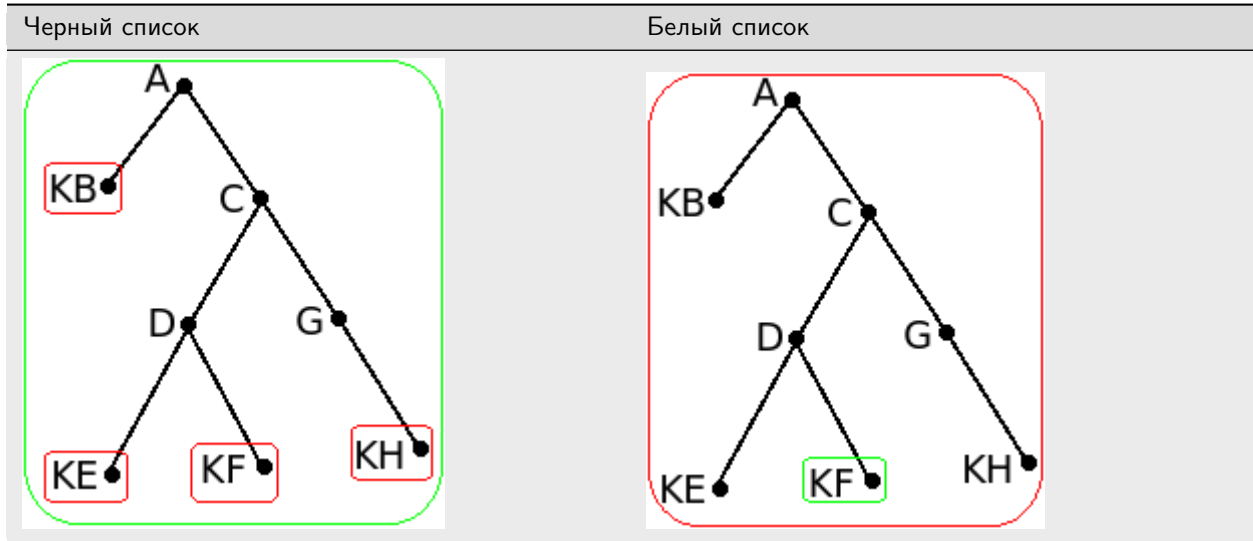
файл черного списка

```
SHOW ALL
HIDE REGEX K
```

файл белого списка

```
HIDE ALL
SHOW CONTENT KF
```

Область действия правила

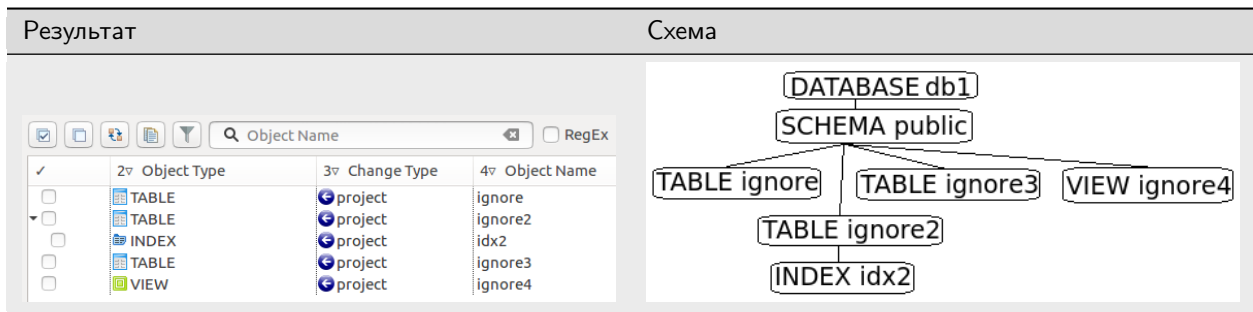


В результате будет отображен объект с названием «KF», т.к. условие белого списка для данного объекта перекрывают по «ширине» условия черного списка.

Примечание: При работе в графической версии pgCodeKeeper добавление второго списка исключений производится путем использования **общего списка исключаемых объектов** или путем добавления внешнего списка через *Хранилище БД. CLI версия* pgCodeKeeper позволяет добавлять дополнительные списки исключений, с помощью команды: `pgcodekeeper-cli --ignore-list <path> SOURCE DEST`.

20.3 Примеры работы с файлом .pgcodekeeperignore

Предположим имеется представление с именем ignore4 и набор из таблиц с именами: ignore, ignore2, ignore3. ignore2 в свою очередь имеет содержимое.



Для того, чтобы исключить все объекты частично попадающие под регулярное выражение «ignore» нужно в .pgcodekeeperignore написать следующие правила:

```
SHOW ALL
HIDE REGEX ignore
```

Результат				Схема																		
☒ ☐ ☐ ☐ ☐ ☐ Object Name ☐ RegEx <table border="1"> <thead> <tr> <th>2v</th> <th>Object Type</th> <th>3v</th> <th>Change Type</th> <th>4v</th> <th>Object Name</th> </tr> </thead> <tbody> <tr> <td>☒</td> <td>TABLE</td> <td>☐ project</td> <td></td> <td></td> <td>ignore2</td> </tr> <tr> <td>☐</td> <td>INDEX</td> <td>☐ project</td> <td></td> <td></td> <td>idx2</td> </tr> </tbody> </table>				2v	Object Type	3v	Change Type	4v	Object Name	☒	TABLE	☐ project			ignore2	☐	INDEX	☐ project			idx2	<pre> graph TD DB[DATABASEdb1] --> SCHEMA[SCHEMA public] SCHEMA --> T1[TABLE ignore] SCHEMA --> T2[TABLE ignore2] SCHEMA --> T3[TABLE ignore3] SCHEMA --> V[VIEW ignore4] T2 --> I[INDEX idx2] </pre>
2v	Object Type	3v	Change Type	4v	Object Name																	
☒	TABLE	☐ project			ignore2																	
☐	INDEX	☐ project			idx2																	

Для того, чтобы исключить объект «ignore2» с содержимым нужно в .pgcodekeeperignore написать следующие правила:

```
SHOW ALL
HIDE CONTENT ignore2
```

Результат				Схема																								
☒ ☐ ☐ ☐ ☐ ☐ Object Name ☐ RegEx <table border="1"> <thead> <tr> <th>2v</th> <th>Object Type</th> <th>3v</th> <th>Change Type</th> <th>4v</th> <th>Object Name</th> </tr> </thead> <tbody> <tr> <td>☐</td> <td>TABLE</td> <td>☐ project</td> <td></td> <td></td> <td>ignore</td> </tr> <tr> <td>☐</td> <td>TABLE</td> <td>☐ project</td> <td></td> <td></td> <td>ignore3</td> </tr> <tr> <td>☐</td> <td>VIEW</td> <td>☐ project</td> <td></td> <td></td> <td>ignore4</td> </tr> </tbody> </table>				2v	Object Type	3v	Change Type	4v	Object Name	☐	TABLE	☐ project			ignore	☐	TABLE	☐ project			ignore3	☐	VIEW	☐ project			ignore4	<pre> graph TD DB[DATABASEdb1] --> SCHEMA[SCHEMA public] SCHEMA --> T1[TABLE ignore] SCHEMA --> T2[TABLE ignore2] SCHEMA --> T3[TABLE ignore3] SCHEMA --> V[VIEW ignore4] T2 --> I[INDEX idx2] </pre>
2v	Object Type	3v	Change Type	4v	Object Name																							
☐	TABLE	☐ project			ignore																							
☐	TABLE	☐ project			ignore3																							
☐	VIEW	☐ project			ignore4																							

Для того, чтобы исключить все объекты с типом «TABLE», частично попадающие под регулярное выражение «ignore» нужно в .pgcodekeeperignore написать следующие правила:

```
SHOW ALL
HIDE REGEX ignore type=TABLE
```

Результат				Схема																								
☒ ☐ ☐ ☐ ☐ ☐ Object Name ☐ RegEx <table border="1"> <thead> <tr> <th>2v</th> <th>Object Type</th> <th>3v</th> <th>Change Type</th> <th>4v</th> <th>Object Name</th> </tr> </thead> <tbody> <tr> <td>☒</td> <td>TABLE</td> <td>☐ project</td> <td></td> <td></td> <td>ignore2</td> </tr> <tr> <td>☐</td> <td>INDEX</td> <td>☐ project</td> <td></td> <td></td> <td>idx2</td> </tr> <tr> <td>☐</td> <td>VIEW</td> <td>☐ project</td> <td></td> <td></td> <td>ignore4</td> </tr> </tbody> </table>				2v	Object Type	3v	Change Type	4v	Object Name	☒	TABLE	☐ project			ignore2	☐	INDEX	☐ project			idx2	☐	VIEW	☐ project			ignore4	<pre> graph TD DB[DATABASEdb1] --> SCHEMA[SCHEMA public] SCHEMA --> T1[TABLE ignore] SCHEMA --> T2[TABLE ignore2] SCHEMA --> T3[TABLE ignore3] SCHEMA --> V[VIEW ignore4] T2 --> I[INDEX idx2] </pre>
2v	Object Type	3v	Change Type	4v	Object Name																							
☒	TABLE	☐ project			ignore2																							
☐	INDEX	☐ project			idx2																							
☐	VIEW	☐ project			ignore4																							

Для того, чтобы исключить все объекты частично попадающие под регулярное выражение «ignore», с типом TABLE и содержимым для указанной базы данных нужно в .pgcodekeeperignore написать следующие правила:

```
SHOW ALL
HIDE CONTENT,REGEX ignore db=name_of_other_db type=TABLE
```

в вышеуказанных правилах использовано название другой базы данных, не той с которой ведется работа в данном примере, поэтому все останется без изменений

Результат				Схема

но если указать название базы данных с которой ведется работа, то из отображаемых объектов исчезнут все объекты кроме одного, который не соответствует типу.

```
SHOW ALL
HIDE CONTENT,REGEX ignore db=db1 type=TABLE
```

Результат				Схема

Для того, чтобы используя черный и белый списки одновременно разрешить отобразить объект «ignore2» нужно написать следующие правила:

файл .pgcodekeeperignore - черный список

```
SHOW ALL
HIDE REGEX ignore
```

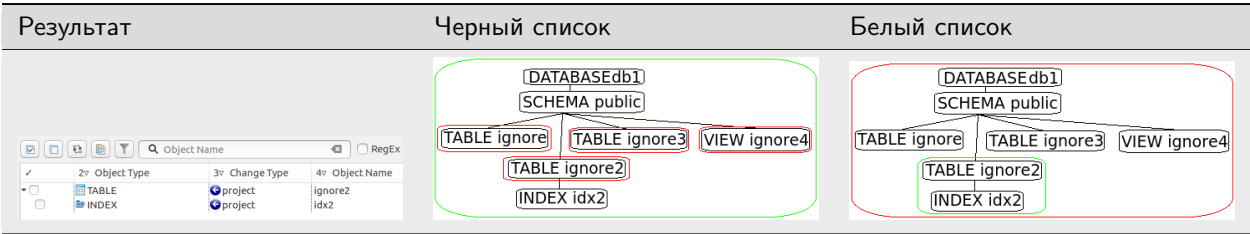
дополнительный файл списка исключаемых объектов - белый список

```
HIDE ALL
SHOW CONTENT ignore2
```

Примечание: Добавление дополнительного списка описано в разделе *Совместное использование черного и белого списков*.

Цель черного списка: исключение всех объектов частично попадающих под регулярное выражение «ignore».

Цель белого списка: убрать из исключенных объектов объект «ignore2».



В результате в сравниваемых объектах останется только объект «ignore2», т.к. благодаря флагу «CONTENT» для объекта «ignore2» правило белого списка перекрывает по «ширине» правило черного списка «HIDE REGEX ignore».

Примечание: Взаимодействие правил разных списков, контролирующих отображение одного и того же объекта, описано в разделе *Совместное использование черного и белого списков*.

Исключение схем при загрузке

В pgCodeKeeper реализована возможность выборочной загрузки объектов из базы данных. Для того, чтобы загрузить содержимое определенных схем, их нужно указать в списке исключаемых схем.

Список исключаемых схем - это список схем с их содержимым, которые не должны или наоборот должны загружаться с БД.

Существует два типа списков исключаемых схем:

черный список

разрешает отображение всех объектов, кроме объектов из указанных схем и самой схемы в правилах списка;

белый список

работает противоположно черному, запрещает отображение всех объектов, кроме объектов из указанных схем и самой схемы в правилах списка.

В каждом проекте можно создать **список исключаемых схем**, распространяющий свое действие только на данный проект, для этого в корне проекта необходимо создать файл **.pgcodekeeperignoreschema** и заполнить его правилами согласно синтаксису списков.

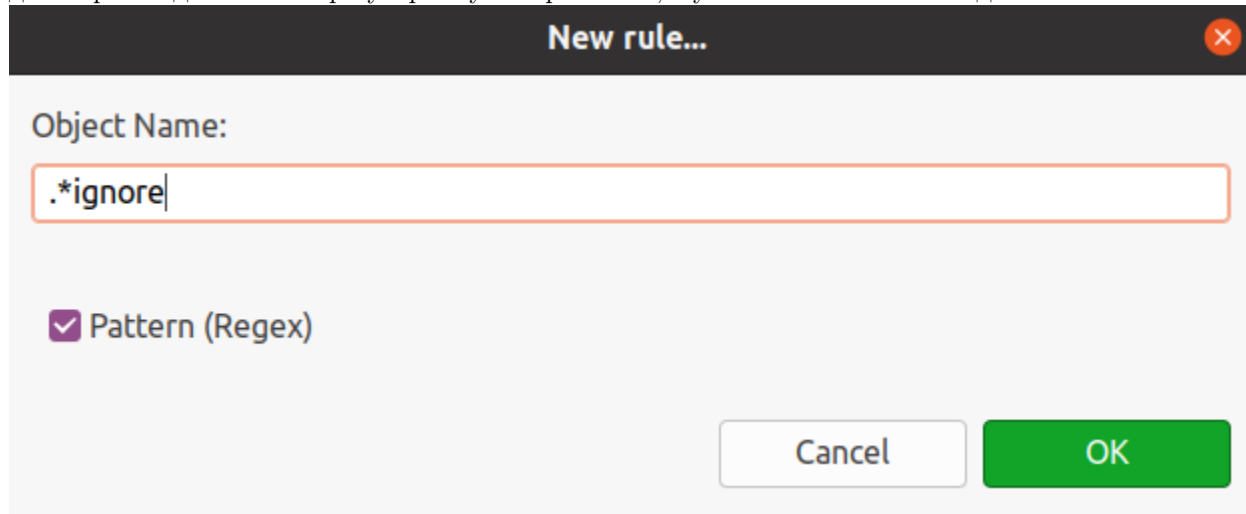
Внимание: Формат файла **.pgcodekeeperignoreschema** чувствителен к регистру. Кодировка должна быть – UTF-8.

Для файла **.pgcodekeeperignoreschema** действуют те же правила заполнения, что и для *Список исключаемых объектов* (смотри раздел *Синтаксис списков*).

Мы можем применить список исключаемых схем и в *Хранилище БД. CLI версия* pgCodeKeeper командой: `pgcodekeeper-cli --ignore-schema <path> SOURCE DEST`.

21.1 Выбор объектов

При выборе флага Pattern(REGEX) в окне создания нового правила поиск объекта будет производиться по регулярному выражению, указанном в поле для имени объекта.



New rule...

Object Name:

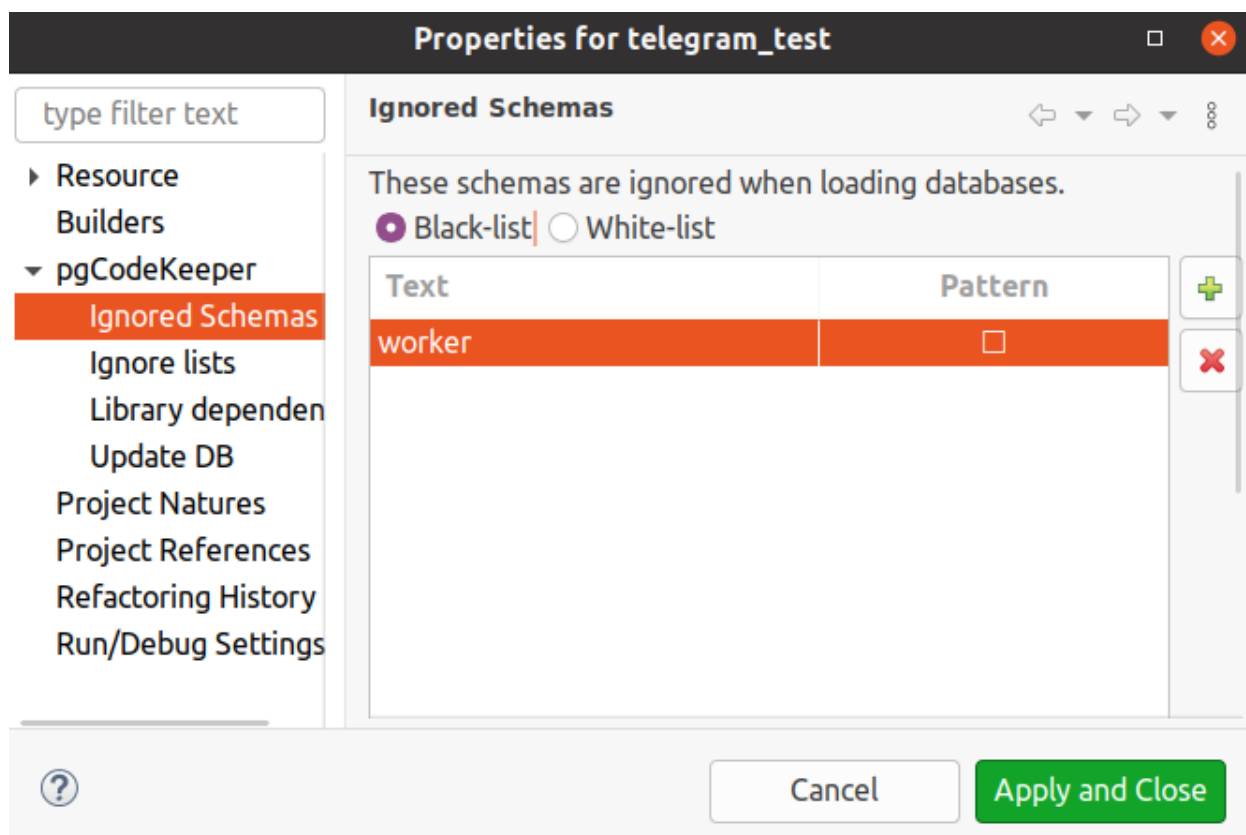
.*ignore|

☒ Pattern (Regex)

Cancel OK

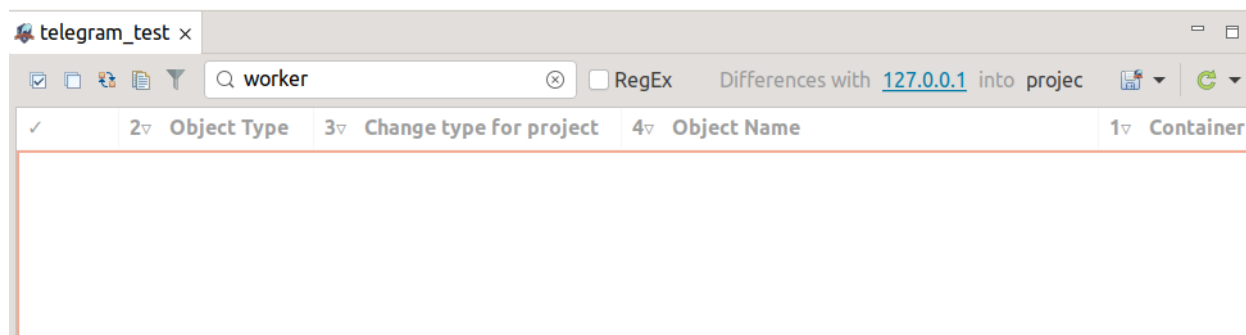
21.2 Пример использования исключения схем при загрузке

Например, нам понадобилось загрузить все объекты из базы данных, кроме объектов из схемы worker. Добавляем в список исключаемых схем worker, выбираем флаг Black-list / Черный список в интерфейсе исключаемых схем:



Нажимаем Apply and Close.

В редакторе проекта кликаем по **Changes** / **Получить изменения** . На панели различий нет объектов из схемы worker!



Если нам необходимо загрузить только объекты из схемы worker, то в интерфейсе исключаемых схем выбираем флаг White-list / Белый список. В результате на панели различий мы получим объекты только из схемы worker.

✓	2▼ Object Type	3▼ Change type for DB	4▼ Object Name	1▼ Container
☐	 SCHEMA	 ALTER	worker	
☐	 FUNCTION	 ALTER	get_changes()	worker
☐	 TABLE	 CREATE	people	worker

В качестве альтернативы плагина для Eclipse IDE предоставляется возможность сравнения схем баз данных через командную строку. Для этого необходимо скачать pgCodeKeeper-cli приложение, которое работает автономно.

Консольная сборка программы может быть использована для:

- создания из дампа файловой структуры «проекта» pgCodeKeeper (без проекта Eclipse)
- создания скриптов наката с использованием дампов и файловых структур pgCodeKeeper
- создания скриптов наката с использованием соединений к базам данных
- просмотра зависимостей объектов

22.1 Начало работы

Для начала необходимо установить [Java SDK](#) для вашей платформы.

Последнюю версию cli сборки можно скачать [здесь](#). В распакованном архиве использовать файл для передачи параметров: **pgcodekeeper-cli.sh** для Linux систем и **pgcodekeeper-cli.bat** для Windows систем.

Набор параметров, доступных для работы с программой Вы можете посмотреть выполнив команду:

```
./pgcodekeeper-cli.sh --help
```

22.2 Примеры

Запись в файл diff.sql скрипт миграции различий из проекта в директории test1 в базу данных testdb1 на сервере localhost:5432:

```
./pgcodekeeper-cli.sh -o /home/codekeeper/projects/diff.sql 'jdbc:postgresql://localhost:5432/testdb1?user=user&password=password' /home/codekeeper/projects/test1/
```

Вывод в консоль скрипт миграции из файла backup в проект в директории test1:

```
./pgcodekeeper-cli.sh /home/codekeeper/backup.sql /home/codekeeper/projects/test1/
```

Создание проекта в директорию test1 на основе базы данных testdb1 на сервере localhost:5432:

```
./pgcodekeeper-cli.sh --parse -o /home/codekeeper/projects/test1/ 'jdbc:postgresql://localhost:5432/testdb1?user=user&password=password'
```

Отображение дерева зависимостей базы данных testdb1 на сервере localhost:5432:

```
./pgcodekeeper-cli.sh --graph 'jdbc:postgresql://localhost:5432/testdb1?user=user&password=password'
```

Отображение дерева объектов, от которых зависят объекты с именами t1 и t2 в базе данных testdb1 на сервере localhost:5432:

```
./pgcodekeeper-cli.sh --graph --graph-reverse --graph-name t1 --graph-name t2 'jdbc:postgresql://localhost:5432/testdb1?user=user&password=password'
```

Все параметры после специального параметра *-vmargs* будут переданы VM.

Использование с ограничениями потребляемой памяти:

```
./pgcodekeeper-cli.sh 1.sql 2.sql -vmargs -Xms256m -Xmx2g
```

Параметр VM *ru.taximaxim.codekeeper.parser.poolsize* позволяет указать количество потоков парсера:

```
./pgcodekeeper-cli.sh 1.sql 2.sql -vmargs -Dru.taximaxim.codekeeper.parser.poolsize=5
```

Windows аутентификация

При работе под ОС Windows для подключения к серверам БД Microsoft SQL доступен режим Windows аутентификации. Это позволяет вместо явного указания имени пользователя и пароля использовать данные текущего пользователя Windows, либо пользователя, от имени которого запущена программа.

Для включения режима Windows аутентификации достаточно включить соответствующую опцию в настройках подключения к БД. В случае использования CLI версии требуется указать параметр `integratedSecurity` в строке подключения.

Примеры использования Windows аутентификации в GUI и CLI:

Create new DB connection

DB Info Ignored objects files Connection properties External loader

DB Host: 127.0.0.1 Port: 5432

DB Name:

DB User:

DB Password:

.pgpass passwords are supported

Read only ☐ disallow executing scripts on DB

Database type MS

MS SQL certificate ☒ trust MS SQL certificate

Domain:

Windows authentication ☒ use Windows authentication for MS SQL Server [Learn more...](#)

Entry Name: 127.0.0.1 ☒ Auto-generate

Group:

Connection Type:

Test connection OK Cancel

```
pgcodekeeper-cli.bat --ms-sql jdbc:sqlserver://127.0.0.1;databaseName={master};
↪integratedSecurity=true NUL
```

23.1 Конфигурация системы

Для использования Windows аутентификации приложению должна быть доступна DLL *mssql-jdbc_auth* Microsoft JDBC драйвера соответствующей версии. Ее следует загрузить из релиза драйвера JDBC и поместить в директорию, присутствующую в переменной окружения *PATH*, либо в свойстве JVM *java.library.path*. Битность библиотеки должна соответствовать битности используемой вами версии Eclipse.

Наиболее простой и универсальный вариант – сохранить библиотеку в директории *C:\Windows\System32*. В 64-битных системах эта директория содержит 64-битные библиотеки. В случае использования 32-битного Eclipse на 64-битной системе, 32-битную библиотеку следует сохранить в директорию *C:\Windows\SysWOW64*.

Вклад участников

В разработке продукта участвовало множество людей. Здесь зафиксирован вклад людей в разработку продукта.

24.1 Идея

- Мокеев Сергей

Хочется отметить, особый вклад в формирование идей, также Новгородова Дмитрия и Володина Андрея.

24.2 Управление

- Саушкин Андрей - управление продуктом и иногда проектом

24.3 Разработка

- Левша Александр
- Галиев Мансур
- Шамсутдинов Ленар
- Ботов Анатолий
- Рябинин Антон
- Акифьев Антон
- Хазиева Гульназ
- Тавтурин Михаил

- Шайдуллин Ильдар
- Сикорский Данил
- Шамсутдинов Эльмир

24.4 Дизайн и вёрстка сайта

- Хазиева Гульназ
- Демаков Василий

24.5 Перевод документации

- Пигин Кирилл
- Мы ищем контрибьюторов для выполнения перевода документации.

Программа pgCodeKeeper разработана и поддерживается ООО «Такстелеком» (ранее ООО «Технология»).

25.1 Карточка организации

Полное наименование	Общество с ограниченной ответственностью «Такстелеком»
Сокращенное наименование	ООО «Такстелеком»
ОГРН	1114501006652
ИНН/КПП	4501170000/720301001
Юридический адрес	640018, Курганская область, г.о. город Курган, г Курган, ул Пролетарская, д. 39, кв. 43
Почтовый адрес	640003, Курганская обл, Курган г, Радионова ул, дом № 17
Телефон	+7 (3522) 63-03-40
Факс	+7 (3522) 60-14-09
ОКПО	87299772
Директор	Татаринцев Игорь Витальевич

25.2 Свидетельство о регистрации

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО
о государственной регистрации программы для ЭВМ
№ 2016611493
pgCodeKeeper

Правообладатель: *Общество с ограниченной ответственностью
«Технология» (RU)*

Авторы: *Левша Александр Александрович (RU), Ботов Анатолий
Вячеславович (RU), Рябинин Антон Владимирович (RU)*

Заявка № **2015661925**
Дата поступления **07 декабря 2015 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ **03 февраля 2016 г.**

Руководитель Федеральной службы
по интеллектуальной собственности



 Г.П. Ивлиев