

# **DataSan. Документация, содержащая информацию, необходимую для эксплуатации экземпляра ПО**

---

Перфоманс Лаб Сервисез



## Содержание

Введение .....	3
Функциональные характеристики.....	3
О документации .....	3
Архитектура .....	5
Минимальные требования к аппаратному и программному обеспечению ..	6
Требования к серверам.....	6
Требования к сетям .....	6
Требования к рабочему месту .....	6
Установка DataSan для PostgreSQL.....	7
Поставка ПО.....	7
Компиляция.....	7
Настройка разрешений .....	9
Профилирование.....	9
Запуск деперсонализации в PostgreSQL .....	13
Логирование.....	14
Решение проблем для PostgreSQL .....	16
Проблемы при установке .....	16
Проблемы при профилировании .....	16
Проблемы при запуске деперсонализации .....	17
Перечень терминов и сокращений.....	18

# Введение

DataSan – инструмент для деперсонализации конфиденциальной информации и персональных данных в БД с сохранением работоспособности системы.

## Функциональные характеристики

DataSan предоставляет пользователю следующие функциональные возможности:

- деперсонализация конфиденциальной информации и персональных данных в базах данных PostgreSQL с сохранением работоспособности системы и консистентности данных;
- защита данных за счет кодирования:
  - хэширование – основной метод кодирования, реализованный на основе метода хэширования MD5;
  - маппинг – метод кодирования, реализованный на основе сопоставления символов;
- добавление функций для обработки:
  - уникальных типов данных, например, ОГРН и ИНН;
  - динамических типов данных, например, транслитерация ФИО;
- добавление функций обработки данных в общий процесс деперсонализации базы данных;
- возобновление деперсонализации базы данных после прерывания процесса, например, при выключении электричества. Деперсонализация продолжится с данных, на которых процесс был прерван;
- многопоточность, которая позволяет использовать инструмент в многоядерных конфигурациях стенда для ускорения процесса деперсонализации;
- очистка вспомогательных объектов по окончании процесса деперсонализации данных;
- встраивание деперсонализации базы данных в процессы разработки.

## О документации

Руководство по эксплуатации предназначено для инженеров нагрузочного и функционального тестирования. После прочтения руководства инженеры смогут:

- установить ПО DataSan;
- запустить деперсонализацию БД;

- проследить за процессом деперсонализации.

Чтобы эффективно использовать DataSan, рекомендуется ознакомиться с работой сервисов и программных продуктов:

- СУБД PostgreSQL;
- среда разработки DBeaver.

Если вам требуется помощь, [свяжитесь с нами](#).

## Архитектура

ПО DataSan представляет собой набор PL/pgSQL-скриптов и Java-классов, которые деперсонализируют конфиденциальную информацию и персональные данные в БД.

Деперсонализированная БД системы удовлетворяет следующим требованиям:

- из БД невозможно получить конфиденциальную информацию о физических или юридических лицах;
- система функционально корректно работает с данной БД.

ПО DataSan состоит из следующих структурных элементов:

Название элемента	Описание
PL/pgSQL-скрипты	Реализуют подготовительные операции с БД, управляют процессом деперсонализации и применением методов кодирования
СУБД PostgreSQL	СУБД на сервере заказчика, для данных которой выполняется деперсонализация
Java-классы	Реализуют методы кодирования для разных типов данных

Соединение выбранных структурных элементов представлено на схеме:



# Минимальные требования к аппаратному и программному обеспечению

## Требования к серверам

Процессор	2 ядра с частотой 2,5 ГГц
Оперативная память	От 8 Гбайт
HDD для IIS и документов	От 50 Гбайт. Зависит от размера хранимых в системе документов
SSD для SQL	От 300 Гбайт
СУБД	PostgreSQL 7 и выше
Установленные расширения для СУБД PostgreSQL	PL/JAVA, pgAgent

## Требования к сетям

Скорость передачи данных	От 10 Мбит/с
--------------------------	--------------

## Требования к рабочему месту

Установленное ПО	DBeaver
------------------	---------

# Установка DataSan для PostgreSQL

Общий порядок установки DataSan для PostgreSQL включает следующие этапы:

1. Перед установкой ознакомьтесь с [типовыми требованиями к аппаратному и программному обеспечению](#).
2. [Получите экземпляр ПО](#).
3. [Скомпилируйте проект](#).
4. [Настройте разрешения для пользователя](#), под которым будет запускаться деперсонализация БД.
5. [Выполните профилирование БД](#).

## Поставка ПО

В рамках поставки клиент получает экземпляр ПО и документацию, которая подробно и понятно описывает эксплуатацию ПО, его настройку, устранение типичных неисправностей и частые вопросы, возникающие у пользователей в процессе эксплуатации.


Экземпляр ПО – 7Z-архив, который содержит файлы для компиляции в среде разработки:

- `1_Create_Java.sql` – алгоритмы обезличивания на языке Java;
- `2_Create_jproc.sql` – методы обезличивания на языке Java;
- `3_Create_Object.txt` – вспомогательные таблицы;
- `4_Pflb_Datasan.txt` – пакет с процедурами DataSan;
- `5.prep proce.txt` – процедуры для создания таблицы **PFLB\_VIEWCONTENT**;
- `6.prep code.txt` – алгоритм для выполнения процедур **PFLB\_VIEWCONTENT**;
- `7_Body.txt` – движок DataSan.

## Компиляция

Чтобы скомпилировать проект:

1. Распакуйте 7Z-архив с экземпляром ПО.
2. Запустите среду разработки, например, DBeaver.
3. В контекстном меню БД выберите пункт **Connect**.
4. Укажите данные для подключения к БД и нажмите на кнопку **Connect**.
5. Последовательно скомпилируйте все файлы из 7Z-архива с экземпляром ПО:
  - А. Откройте файл для компиляции и скопируйте его содержимое.
  - В. В IDE в область **Worksheet** вставьте содержимое файла.

С. Выделите текст с помощью горячих клавиш **CTRL+A** и нажмите на кнопку .

После компиляции структура проекта выглядит следующим образом:

- Tables:
  - PFLB\_ACTIVE\_JOBS;
  - PFLB\_ACTIVE\_STATUS;
  - PFLB\_ALL\_TAB\_INDEXES;
  - PFLB\_ALL\_TRIGGERS;
  - PFLB\_CURRENT\_STATUS;
  - PFLB\_EVEN\_TABLES;
  - PFLB\_JOB\_PROCESSED\_QUERIES;
  - PFLB\_LOGS;
  - PFLB\_LT\_DEPERSONTABLES;
  - PFLB\_NOT\_PROCESSED\_JOBS;
  - PFLB\_PROCESSED\_QUERIES;
  - PFLB\_TABLE\_SIZE;
  - PFLB\_TABLE\_TRIGGERS;
  - PFLB\_TEMP\_TABLE\_INDEXES;
  - PFLB\_VIEWCONTENT;
- Packages:
  - PFLB\_DATASAN;
  - PFLB\_DATASAN\_PREP;
- Functions:
  - PFLB\_ENCODE\_HASH\_BDATE;
  - PFLB\_ENCODE\_HASH\_BINARY;
  - PFLB\_ENCODE\_HASH\_CARDNUMBER;
  - PFLB\_ENCODE\_HASH\_CDATE;
  - PFLB\_ENCODE\_HASH\_CHAR;
  - PFLB\_ENCODE\_HASH\_CHARDEL;
  - PFLB\_ENCODE\_HASH\_DATE;
  - PFLB\_ENCODE\_HASH\_INNUMBER;
  - PFLB\_ENCODE\_HASH\_IP;
  - PFLB\_ENCODE\_HASH\_NCHAR;
  - PFLB\_ENCODE\_HASH\_OGRN;
  - PFLB\_ENCODE\_HASH\_PHONENUMBER;
  - PFLB\_ENCODE\_HASH\_PPCHAR;
  - PFLB\_ENCODE\_HASH\_SNILSNUMBER;
- Java:
  - BitConverter;
  - OraSQL.



## Настройка разрешений

Настройте разрешения для пользователя, под которым будет запускаться деперсонализация БД:

```
GRANT ALL on schema PFLBTEST to PFLBUSER;  
GRANT ALL ON ALL SEQUENCES IN SCHEMA PFLBTEST TO PFLBUSER;  
GRANT USAGE ON LANGUAGE java TO PFLBUSER;
```

## Профилирование

Профилирование – поиск колонок в таблицах в базе данных, содержащих персональные данные.

Чтобы выполнить профилирование БД:

1. Найдите таблицы и поля в БД, которые содержат персональные данные:
  - A. Получите список всех таблиц, соответствующих определенной схеме:

```
SELECT schemaname owner,relname table_name,n_live_tup num_rows  
FROM pg_stat_user_tables  
where schemaname='schema_name'  
ORDER BY n_live_tup DESC;
```

Пример списка таблиц:

	ABC OWNER ▼	ABC TABLE_NAME ▼	123 NUM_ROWS ↓▼
1	USERFORDEPERS	DATASAN_EMPLOYEES	300 001
2	USERFORDEPERS	DATASAN_ACCOUNT	300 001
3	USERFORDEPERS	CLIENTS_BACKUP	300 001
4	USERFORDEPERS	DATASAN_CLIENTS	300 001
5	USERFORDEPERS	PFLB_LOGS	38
6	USERFORDEPERS	DATASAN_BLACKLIST	8
7	USERFORDEPERS	DATASAN_SUPPLIERS	7
8	USERFORDEPERS	DATASAN_DELIVERY	6
9	USERFORDEPERS	PFLB_VIEWCONTENT	5
10	USERFORDEPERS	DATASAN_PRODUCT	5

- B. Выберите таблицы с большим количеством строк. Рекомендуется выбирать таблицы, которые содержат более 10000 строк.
  - C. Посмотрите содержимое каждой из найденных таблиц.
  - D. Сохраните тип данных и названия полей, которые содержат персональные данные.
2. Соберите представления на основании найденных полей и выберите методы кодирования для каждого типа данных. Метод кодирования

указывается в конце наименования представления:

- **char** – смешанные и неопределенные поля;
- **chardel** – адреса и ФИО;
- **nchar** – названия организации и ФИО;
- **phonenumber** – номера телефонов;
- **ppchar** – серия и номер паспорта;
- **bdate** – дата рождения;
- **inn\_umber** – ИНН;
- **cdate** – даты в интервале за год до текущего дня;
- **cardnumber** – номер банковской карты;
- **ip** – IP-адреса.

Примеры:

- создание представления для поля **%E-MAIL%** и кодирование методом **char**:

```
create view PFLB_LT_Deperson_CommonEmail_CHAR
as
select C.*
from information_schema.columns c
inner join information_schema.tables t on c.table_name=t.table_name
where
    (C.COLUMN_NAME ILIKE '%EMAIL%' or C.COLUMN_NAME ILIKE '%E_MAIL%'
or C.COLUMN_NAME ILIKE '%E-MAIL%')
and C.DATA_TYPE = 'character varying';
```

- создание представления для поля **%INN%** и кодирование методом **inn\_umber**:

```
create view PFLB_LT_Deperson_Common_INNNUMBER
as
select C.*
from information_schema.columns c
inner join information_schema.tables t on c.table_name=t.table_name
where
    C.COLUMN_NAME ILIKE '%INN%'
and (C.DATA_TYPE = 'integer' or C.DATA_TYPE = 'numeric');
```

- создание представления для полей с названиями организаций и ФИО, их кодирование методом **nchar**:

```
create view PFLB_LT_Deperson_PhioFIO_NCHAR
as
select C.*
from information_schema.columns c
inner join information_schema.tables t on c.table_name=t.table_name
where
    (c.table_name not like 'pflb%' and c.table_name not like
'viewcontent')
```

```

and
(
  C.COLUMN_NAME ILIKE '%FIRST_NAME%'
or C.COLUMN_NAME ILIKE '%LAST_NAME%'
or C.COLUMN_NAME ILIKE '%SECOND_NAME%'
or C.COLUMN_NAME ILIKE '%MIDDLENAME%'
or C.COLUMN_NAME ILIKE '%PAYERNAME%'
or C.COLUMN_NAME ILIKE '%RECIEVERNAME%'
or C.COLUMN_NAME ILIKE '%CLIENTFULLNAME%'
or C.COLUMN_NAME ILIKE '%PERSONFULLNAME%'
or C.COLUMN_NAME ILIKE '%PAYERFULLNAME%'
or C.COLUMN_NAME ILIKE '%RECIEVERFULLNAME%'
or C.COLUMN_NAME ILIKE '%BUYERFULLNAME%'
or C.COLUMN_NAME ILIKE '%SELLERFULLNAME%'
or C.COLUMN_NAME ILIKE '%BENEFICAIARYFULLNAME%'
or C.COLUMN_NAME ILIKE '%BOSSFULLNAME%'
or C.COLUMN_NAME ILIKE '%KINFULLNAME%'
or C.COLUMN_NAME ILIKE '%NAME%'
)
and C.DATA_TYPE = 'character varying'
and C.TABLE_SCHEMA != 'information_schema';

```

3. Убедитесь, что поля в представлениях не повторяются, иначе в таблице для деперсонализации данных будут дублироваться строки.
4. Вставьте в промежуточную таблицу **PFLB\_LT\_DepersonTables** содержимое всех представлений. Например:

```

insert into PFLB_LT_DepersonTables values
('PFLB_LT_Deperson_CommonEmail_CHAR');
insert into PFLB_LT_DepersonTables values
('PFLB_LT_Deperson_Common_INNNUMBER');
insert into PFLB_LT_DepersonTables values
('PFLB_LT_Deperson_PhisFIO_NCHAR');

```

5. Запустите процедуру по наполнению основной таблицы для деперсонализации данных **PFLB\_VIEWCONTENT**:

```

CALL PFLB_DATASAN_PREP.PFLB_FILL_VIEWCONTENT('HASH');

```

Таблица **PFLB\_VIEWCONTENT** наполнится записями вида **имя\_таблицы + наименование\_колонки**, которые ранее были собраны в представлениях.

Пример наполненной таблицы **PFLB\_VIEWCONTENT**:

	OWNER_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_TYPE	ENCODE_METHOD	COLUMN_ENCODE_TYPE	EXAMPLE
1	TEST2_DB	DATASAN_CLIENTS	CLIENT_ADDRESS	VARCHAR	HASH	CHARDEL	554039, Тюменская область, город Н
2	TEST2_DB	DATASAN_CLIENTS	CLIENT_FULLNAME	VARCHAR	HASH	CHARDEL	Лихачев Владимир Христофорович
3	TEST2_DB	DATASAN_CLIENTS	CLIENT_PHONE_NUMBER	VARCHAR	HASH	PHONENUMBER	7(477)196-49-67
4	TEST2_DB	DATASAN_CLIENTS	CLIENT_PASSPORT	VARCHAR	HASH	PPCHAR	0123 456789
5	TEST2_DB	DATASAN_CLIENTS	CLIENT_BIRTHDAY	DATE	HASH	BDATE	01.01.2021

6. Проанализируйте каждую строку таблицы **PFLB\_VIEWCONTENT** и при необходимости скорректируйте ее:
  - удалите все дубликаты, неправильно выбранные методы, пустые или системные таблицы. Например:

OWNER_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_TYPE	ENCODE_METHOD	COLUMN_ENCODE_TYPE	EXAMPLE
TEST2_DB	DATASAN_CLIENTS	CLIENT_ADDRESS	VARCHAR	HASH	CHARDEL	554039, Тюменская область, город Т
TEST2_DB	DATASAN_CLIENTS	CLIENT_FULLNAME	VARCHAR	HASH	CHARDEL	/Лихачёв Владимир Христофорович
TEST2_DB	DATASAN_CLIENTS	CLIENT_PHONE_NUMBER	VARCHAR	HASH	PHONENUMBER	7(477)196-49-67
TEST2_DB	DATASAN_CLIENTS	CLIENT_PASSPORT	VARCHAR	HASH	PPCHAR	0123 456789
TEST2_DB	DATASAN_CLIENTS	CLIENT_BIRTHDAY	DATE	HASH	BDATE	01.01.2021
TEST2_DB	DATASAN_CLIENTS	CLIENT_BIRTHDAY	DATE	HASH	BDATE	01.01.2021

- удалите таблицы, которые не содержат персональные данные.  
Например, таблица **DATASAN\_LOGS** содержит колонку  
CLIENT\_ADDRESS, но в этом поле нет персональных данных:

6	USERFORDEPERS	DATASAN_LOGS	CLIENT_ADDRESS	VARCHAR	HASH	CHARDEL	Случайный комент
---	---------------	--------------	----------------	---------	------	---------	------------------

# Запуск деперсонализации в PostgreSQL

Чтобы запустить деперсонализацию БД, выполните процедуру

`PFLB_DATASAN.PFLB_PROCESS_DATA_TYPE:`

```
CALL PFLB_DATASAN.PFLB_PROCESS_DATA_TYPE('Лицензионный ключ', 'Метод кодирования по умолчанию', 'Количество потоков', 'Количество строк', 'Режим работы');
```

Параметры процедуры:

- **Лицензионный ключ.** Ключ, который получает клиент после покупки ПО DataSan;
- **Метод кодирования по умолчанию.** Используется, если в таблице **PFLB\_VIEWCONTENT** не указан метод кодирования. Возможные значения:
  - **HASH.** Хэширование – основной метод кодирования, реализованный на основе метода хэширования MD5;
  - **MAPPING.** Маппинг – метод кодирования, реализованный на основе сопоставления символов;
- **Количество потоков.** Количество потоков деперсонализации, запускаемых параллельно. Если ресурсы на сервере ограничены, укажите количество потоков равное количеству ядер;
- **Количество строк.** Количество строк таблицы для обработки внутри одного потока. Рекомендуется указать не менее 50000 и не более 300000;
- **Режим работы.** Возможные значения: **0** – обычный режим, **1** – режим отладки.

Пример запуска процедуры:

```
CALL PFLB_DATASAN.PFLB_PROCESS_DATA_TYPE((( '1ACB-2387BD-4592', 'HASH', 64, 300000, 0);
```

Чтобы посмотреть список всех обезличенных таблиц, выполните запрос после завершения процесса деперсонализации:

```
SELECT *FROM PFLB_PROCESSED_QUERIES
```

После проведения деперсонализации удостоверьтесь, что все таблицы обезличены:

	CLIENT_ID	CLIENT_ID_ACCOUNT	CLIENT_ADDRESS	CLIENT_FULLNAME	CLIENT_PHONE_NUMBER	CLIENT_PASSPORT	CLIENT_BIRTHDAY
1	35	300036760168	Бекхенасмафн шостыт, езэлк Ээруан-Кычч, вк. Цымпыбэв, 08	Юлиб Хужарат Эбчуйосык	7(713)486-14-73	0349706407	01.01.14
2	36	300037847634	Гэшоф шостыт, езэлк Шофес, нч. Луифач, 94	Шоковдп Кэкса Ифкыкык	7(491)446-77-60	8308067869	01.09.70
3	37	300038831297	Икляшшэцэб шостыт, езэлк Евитблх, кяпп Цымпыбэв, 15	Тнахицу Сыншиэ Врийгышшо	7(301)748-82-84	7332099107	01.06.00
4	38	300039230041	Бекхенасмафн шостыт, езэлк Суруойшэжт, кэв. Рыччэв, 14	Эзэв Шэффис Цэлибэй	7(541)222-92-84	6148606650	01.09.70
5	39	300040462633	Гчэжэошдл шостыт, езэлк Пычк, азыбди Зуыйщрибуш, 15	Юлиб Хужарат Эбчуйосык	7(198)411-32-47	2454704058	01.01.14
6	40	300041915154	Ечруоифи шостыт, езэлк Натчэфйяк, омытэ Добто, 72	Нйкетбш Ытэупы Кэуфосыфч	7(615)324-69-51	3859055010	01.01.14
7	41	300042915154	Ечруоифи шостыт, езэлк Натчэфйяк, омытэ Добто, 72	Нйчэвэвтай Кэбэшис Йсардэо	7(043)699-01-15	3170797115	01.06.00
8	42	300043369846	Иомэгошдл шостыт, езэлк Езэв-Цойлш, эедлх Зуыйщрибуш, 08	Етэти Пойлиту Сэбэсэмык	7(166)285-62-07	1837437685	01.09.70
9	43	300044811815	Цэфиней шостыт, езэлк Асэнт, эедлх Цымпыбэв, 77	Пэзэсц Пхожарыб Цэвэкелтэ	7(335)744-24-16	4289747045	01.09.70
10	44	300045495320	Ышвонжук шостыт, езэлк Илчэнтэб, кел. Изйерэрицэв, 67	Ыжлшйшэв Нцгину Гтэвмчэв	7(541)222-92-84	2949189395	01.09.70
11	45	300046084992	Хродэвэвфус шостыт, езэлк Буэвнч, кэвэл Добто, 85	Почувнэк Слтууак Нпшэуфч	7(003)063-80-96	2556209876	01.09.70
12	46	300047323599	Ымплэтат шостыт, езэлк Шыкык, омытэ Цымпыбэв, 47	Унэвэвк Йнэвэв Шобчуучч	7(741)768-86-02	1657541619	01.09.70
13	47	300048831297	Икляшшэцэб шостыт, езэлк Евитблх, кяпп Цымпыбэв, 15	Тнахицу Сыншиэ Врийгышшо	7(038)014-35-37	1679707336	01.06.00

Note:

Данные после обезличивания методом хэширования не подлежат восстановлению.

## Логирование

После запуска деперсонализации можно посмотреть общий лог процесса, статус и активные задания.

### Общий лог деперсонализации

Чтобы посмотреть лог всего процесса, выполните запрос:

```
SELECT * FROM PFLB_logs ORDER BY id desc
```

В таблице **PFLB\_logs** отображается текущий этап и таблица:

```
10 !@#$ Starting to update DATASAN_CLIENTS
9 !@#$ Disable indexes for DATASAN_CLIENTS
8 #index table is filled
7 Update string is created
6 clear #index table
5 !@#$ Generating update string for DATASAN_CLIENTS
4 !@#$ Start processing on DATASAN_CLIENTS
3 !@#$ Lic key is verificated
2 !@#$ enter to generate_update
```

Когда в таблице **PFLB\_logs** появляются сообщения **All temp columns dropped** и **exit from generate\_update**, процесс деперсонализации завершился, и все таблицы из **PFLB\_VIEWCONTENT** обезличены:

```
39 !@#$ exit from generate_update
38 !@#$ All temp columns dropped
37 Checking temp_id_column on USERFORDEPERS.DATASAN_CLIENTS, start dropping it
36 !@#$ Start processing on DATASAN_CLIENTS
35 !@#$ Lic key is verificated
```

## Активные задания

Чтобы посмотреть активные задания, выполните запрос:

```
SELECT * FROM PFLB_ACTIVE_JOBS
```

Если задания в таблице **PFLB\_ACTIVE\_JOBS** не меняются долгое время, то процесс деперсонализации остановился с ошибкой.

## Статус процесса

Чтобы посмотреть статус процесса, выполните запрос:

```
SELECT * FROM PFLB_ACTIVE_STATUS
```

Если по завершению процесса таблица **PFLB\_ACTIVE\_STATUS** пуста, то процесс завершился без ошибок.

Если в таблице **PFLB\_ACTIVE\_STATUS** есть поле ERROR, то процесс завершился с ошибками. Посмотрите содержимое таблицы **PFLB\_logs**. Подробнее см. раздел [Решение проблем](#).

# Решение проблем для PostgreSQL

## Проблемы при установке

### Недостаточно разрешений

Отсутствуют необходимые разрешения у пользователя, под которым запускается деперсонализация БД.

#### Решение:

Запросите необходимые разрешения у администратора БД. Подробнее см. раздел [Настройка разрешений](#).

### Таблица или представление пользователя отсутствует

Не были скомпилированы некоторые таблицы, представления или процедуры.

#### Решение:

1. Найдите, какие таблицы, представления или процедуры не были скомпилированы из списка всех вспомогательных таблиц.
2. Скомпилируйте необходимые таблицы, представления или процедуры повторно.

Подробнее см. раздел [Компиляция](#).

## Проблемы при профилировании

### Неправильно выбран метод обезличивания данных

#### Решение:

Замените метод обезличивания через **update** в таблице **PFLB\_VIEWCONTENT** на необходимый метод из списка методов. Подробнее см. раздел [Профилирование](#).

### Дубликаты в таблице PFLB\_VIEWCONTENT

#### Решение:

1. Проверьте содержимое представлений и корректность списка колонок. Некоторые колонки дублируются или агрегируются неправильно из-за символов **%%** в представлении.
2. Явно укажите название колонки или укажите название с помощью символов **%%**, игнорируя символы до и после.
3. Удалите похожие колонки.

### В таблицу PFLB\_VIEWCONTENT попали колонки без



## персональных данных

### Возможные причины и решения:

- **PFLB\_VIEWCONTENT** – промежуточная таблица. На этапе профилирования в ней могли находиться персональные данные, но спустя время они были удалены. Рекомендуется оставить колонки.
- Колонки попали при создании представлений и таблицы **PFLB\_VIEWCONTENT**. Проверьте колонки вручную и при необходимости удалите их.

## Нет метода для определенного вида персональных данных

### Решение:

Обратитесь в службу техподдержки ООО «Перфоманс Лаб Сервисез» по электронной почте [datasan@pflb.ru](mailto:datasan@pflb.ru).

## Проблемы при запуске деперсонализации

### Дублируется имя столбца

#### Решение:

1. К таблице **PFLB\_VIEWCONTENT** выполните запрос с помощью оператора DISTINCT.
2. Удалите все дубликаты.

### Неверное имя таблицы

#### Решение:

Проверьте правильность заполнения таблицы **PFLB\_VIEWCONTENT** и найдите ошибку в имени таблицы.

# Перечень терминов и сокращений

## **БД**

База данных

## **Деперсонализация БД**

Обезличивание конфиденциальной информации в базе данных с сохранением работоспособности системы

## **Деперсонализация данных**

Общее название любого процесса удаления связи между совокупностью идентифицирующих данных и субъектом данных

## **ИБ**

Информационная безопасность

## **Обезличивание**

Действия, в результате которых становится невозможным без использования дополнительной информации определить принадлежность персональных данных конкретному субъекту персональных данных (ФЗ N 152 "О персональных данных" от 27.07.2006, статья 3 пункт 9)

## **Персональные (личные) данные**

Любая информация, относящаяся к прямо или косвенно определенному или определяемому физическому лицу (субъекту персональных данных) (ФЗ N 152 "О персональных данных" от 27.07.2006, статья 3 пункт 1)

## **ПО**

Программное обеспечение

## **Профилирование**

Поиск колонок в таблицах в базе данных, содержащих персональные данные.

## **СУБД**

Система управления базой данных