**ODBC log / trace parser**

Парсер предназначен для извлечения SQL-запросов и автоматической подстановки параметров из ODBC-лога или трассы. Для работы нужно скопировать фрагмент лога/трассы в файл log.txt и запустить скрипт (необходим Python 3.11): py process\_log.py   
Для проверки версии, установки Python и запуска парсера можно запустить start.bat

Парсер работает с двумя форматами:

### Формат ODBC-лога:

Строки начинаются с ‘[’, блок содержит строки с маркерами:

[2023-Nov-17 17:13:54.323608] <NOTE> **NSQL**:  исходный запрос, который приходит из прикладного кода. (NATIVE\_SQL)

[2023-Nov-17 17:13:54.408382] <NOTE> **PHSQL**:  запрос, обработанный конвертером, там пронумерованы placeholder. (PLACEHOLDER\_SQL)

[2023-Nov-17 17:13:54.416360] <NOTE> **bind map**:  карта placeholder

[2023-Nov-17 17:13:54.416360] <NOTE> **SQL**:  запрос pgsql из конвертера, обработанный “перестановщиком placeholder”, все placeholder заменены на ‘?’

[2023-Nov-17 17:13:54.417357] <NOTE> **to RECORD**:  строки с параметрами.

### Формат трассы:

Первая строка трассы начинается с любого символа, кроме ‘[’. Для парсинга Oracle-трассы следует на первой строке входного файла log.txt поставить строку 'ORACLE'. Расположение полей в трассе у разных пользователей может различаться, в связи с этим может потребоваться доработка парсера.

Пример форматов трассы, с которой работает эта реализация.

#1, type RSDLONG, value: 1 RsdCmdSp.cpp (line 1710) Parameter\Input 4 RSDODBC 12:22:45.955 20-11-2023

или:

RSDODBC Parameter\Input #1, type RSDLONG, value: 505 4 11:40:27.029 23-11-2023 RsdCmdSp.cpp (line 1710)

Строка с параметром определяется по наличию маркера 'Parameter'

#1, type RSDSHORT, value: -32768 RsdCmdSp.cpp (line 1561) Parameter\Input

Тело запроса считывается со строки, следующей за последним параметром:

RSDODBC Parameter\Input #15, type RSDDATE, value: 8.12.2025 4   
RSDODBC SQL\Execute select t.t\_infoid, t.t\_direct, t.t\_trn, t.t\_r 1

Запрос считывается до строки, содержащей Result=

RSDODBC SQL\Execute Result=

При парсинге трассы скрипт обрабатывает все встреченные блоки #1 ... Result= , последовательно сохраняя все найденные запросы в выходной файл.

### Вывод результатов:

Запросы с параметрами сохраняются в файлы req\_OR\_PARAMS.sql / req\_PG\_PARAMS.sql / req\_TRACE\_PARAMS.sql в папке со скриптом.

#### Примечание:

В зависимости от формата вывода трассы может потребоваться в функции **find\_req\_trace()** подобрать параметр **shift**. По умолчанию заданы 2 значения - 3 и 7, соответствующие приведённым выше примерам трассы. Переключение между ними происходит в строке:

pos\_end = len(line) - len(\_end) - shift[0]