



README: Система отчетов КСК

Оглавление

1. [Обзор системы](#)
2. [Архитектура](#)
3. [Доступные отчеты](#)
4. [Быстрый старт](#)
5. [Функции генерации отчетов](#)
6. [Просмотр результатов](#)
7. [Автоматизация](#)
8. [Управление жизненным циклом](#)
9. [Troubleshooting](#)

Обзор системы

Система отчетов КСК предоставляет полный набор инструментов для генерации, хранения и управления отчетами по контролю санкционных списков.

Основные возможности

- **5 типов отчетов** - от общей статистики до детальных данных по фигурантам
- **Автоматическое управление TTL** - системные отчеты хранятся 365 дней, пользовательские 7-14 дней
- **Системное логирование** - все операции записываются в `ksk_system_operations_log`
- **Фильтрация данных** - поддержка параметров через JSONB
- **Оптимизированные запросы** - использование структурированных полей вместо JSON

Архитектура

Таблицы системы

ksk_report_orchestrator	- Метаданные типов отчетов
ksk_report_header	- Заголовки созданных отчетов
ksk_report_totals_data	- Данные: общая статистика
ksk_report_list_totals_data	- Данные: итоги по спискам
ksk_report_totals_by_payment_type_data	- Данные: статистика по типам платежей
ksk_report_list_totals_by_payment_type_data	- Данные: итоги по спискам и типам
ksk_report_figurants_data	- Данные: статистика по фигурантам
ksk_system_operations_log	- Системный лог операций

Типы платежей

Система поддерживает 5 типов платежей (русские названия):

- **Входящий** - входящие платежи
- **Исходящий** - исходящие платежи
- **Транзитный** - транзитные операции
- **Межфилиальный** - межфилиальные переводы
- **Внутрифилиальный** - внутрифилиальные операции

Доступные отчеты

Код отчета	Название	Описание	TTL (system/user)
totals	Общая статистика	Подсчет транзакций по резолюциям	365/14 дней
list_totals	Итоги по спискам	Агрегация по кодам санкционных списков	365/14 дней
totals_by_payment_type	Статистика по типам платежей	Разбивка по 5 типам платежей	365/14 дней
list_totals_by_payment_type	Итоги по спискам и типам	Комбинированная агрегация	365/14 дней
figurants	Отчет по фигурантам	Детальная статистика по каждому фигуранту	30/7 дней

Быстрый старт

Создание первого отчета

```
-- Создать отчет по общей статистике за вчера
SELECT ksk_run_report(
    'totals',          -- Код отчета
    'system',          -- Инициатор
    NULL,              -- user_login (NULL для system)
    CURRENT_DATE - 1, -- Начальная дата
    CURRENT_DATE - 1, -- Конечная дата
    NULL               -- Параметры
);
-- Результат: ID созданного отчета (например, 42)
```

Просмотр результатов

```
-- Найти созданный отчет
SELECT
    id,
    name,
    status,
    created_datetime,
    finished_datetime
FROM ksk_report_header
WHERE id = 42;

-- Посмотреть данные отчета
SELECT *
FROM ksk_report_totals_data
WHERE report_header_id = 42;
```

Функции генерации отчетов

1. `ksk_run_report()` - Универсальная функция запуска

Назначение: Создает и запускает отчет любого типа

Синтаксис:

```
ksk_run_report(
    p_report_code     VARCHAR,      -- Код отчета из оркестратора
    p_initiator       VARCHAR,      -- 'system' или 'user'
    p_user_login      VARCHAR,      -- Логин пользователя (NULL для system)
    p_start_date      DATE,        -- Начальная дата периода
    p_end_date        DATE,        -- Конечная дата (NULL = start_date)
    p_parameters      JSONB,       -- Дополнительные параметры
)
RETURNS INTEGER -- ID созданного отчета
```

Примеры:

```
-- 1. Системный отчет за день
SELECT ksk_run_report('totals', 'system', NULL, '2025-10-22', NULL, NULL);

-- 2. Пользовательский отчет за неделю
SELECT ksk_run_report(
    'list_totals',
    'user',
    'ivanov',
    '2025-10-15',
    '2025-10-22',
    NULL
);

-- 3. Отчет по фигурантам с фильтром по спискам
SELECT ksk_run_report(
    'figurants',
    'user',
    'petrov',
    '2025-10-20',
    '2025-10-22',
    '{"list_codes": ["4200", "4204"]}'::JSONB
);

-- 4. Отчет за месяц
SELECT ksk_run_report(
    'totals_by_payment_type',
    'system',
    NULL,
    '2025-10-01',
    '2025-10-31',
    NULL
);
```

2. Отчет: Общая статистика (totals)

Что включает:

- Всего обработано транзакций
- Транзакции без результата (resolution='empty')
- Транзакции с результатом
- Разбивка по резолюциям: allow, review, deny
- Количество bypass

Пример:

```
-- Создать отчет за вчера
SELECT ksk_run_report('totals', 'system', NULL, CURRENT_DATE - 1, NULL, NULL);

-- Посмотреть результат
```

```

SELECT
    total,
    total_without_result,
    total_with_result,
    total_allow,
    total_review,
    total_deny,
    total_bypass
FROM ksk_report_totals_data
WHERE report_header_id = (
    SELECT id FROM ksk_report_header
    WHERE name LIKE 'Общая статистика%'
    ORDER BY created_datetime DESC
    LIMIT 1
);

```

Результат:

total	total_without_result	total_with_result	total_allow	total_review	total_deny
50000	33000	17000	15000	1800	200

3. Отчет: Итоги по спискам (list_totals)

Что включает:

- Одна строка на каждый код санкционного списка
- Счётчики по резолюциям для каждого списка

Пример:

```

-- Создать отчет
SELECT ksk_run_report('list_totals', 'system', NULL, '2025-10-22', NULL, NULL);

-- Посмотреть результат
SELECT
    list_code,
    total_with_list,
    total_allow,
    total_review,
    total_deny,
    total_bypass
FROM ksk_report_list_totals_data
WHERE report_header_id = (SELECT MAX(id) FROM ksk_report_header)
ORDER BY total_with_list DESC
LIMIT 10;

```

Результат:

list_code	total_with_list	total_allow	total_review	total_deny	total_bypass
-----------	-----------------	-------------	--------------	------------	--------------

4200		5000		4500		450		50		30
4204		3000		2700		280		20		15
4210		2500		2300		180		20		10

4. Отчет: Статистика по типам платежей (totals_by_payment_type)

Что включает:

- Общие счётчики
- Разбивка по каждому из 5 типов платежей

Пример:

```
-- Создать отчет
SELECT ksk_run_report(
    'totals_by_payment_type',
    'system',
    NULL,
    '2025-10-01',
    '2025-10-31',
    NULL
);

-- Посмотреть результат
SELECT
    total,
    i_total AS входящий,
    o_total AS исходящий,
    t_total AS транзитный,
    m_total AS межфилиальный,
    v_total AS внутрифилиальный,
    i_total_review AS входящий_review,
    o_total_review AS исходящий_review
FROM ksk_report_totals_by_payment_type_data
WHERE report_header_id = (SELECT MAX(id) FROM ksk_report_header);
```

5. Отчет: Итоги по спискам и типам (list_totals_by_payment_type)

Что включает:

- Комбинация list_code и типов платежей
- Одна строка на каждый list_code со счётчиками по всем типам

Пример:

```
SELECT ksk_run_report(
    'list_totals_by_payment_type',
    'user',
    'analyst',
    '2025-10-22',
```

```
    NULL,  
    NULL  
);
```

6. Отчет: Статистика по фигурантам (figurants)

Что включает:

- Детальная информация по каждому фигуранту
- Опциональная фильтрация по кодам списков

Пример без фильтра:

```
-- Все фигуранты  
SELECT ksk_run_report(  
    'figurants',  
    'user',  
    'compliance_officer',  
    '2025-10-22',  
    NULL,  
    NULL  
);  
  
-- Посмотреть результат  
SELECT  
    list_code,  
    name_figurant,  
    president_group,  
    total,  
    total_review,  
    total_deny,  
    exclusion_phrase  
FROM ksk_report_figurants_data  
WHERE report_header_id = (SELECT MAX(id) FROM ksk_report_header)  
ORDER BY total DESC  
LIMIT 20;
```

Пример с фильтром по спискам:

```
-- Только списки 4200 и 4204  
SELECT ksk_run_report(  
    'figurants',  
    'user',  
    'analyst',  
    '2025-10-20',  
    '2025-10-22',  
    '{"list_codes": ["4200", "4204"]}'::JSONB  
);
```

Просмотр результатов

Список всех отчетов

```
SELECT
    h.id,
    o.name AS report_type,
    h.name AS report_name,
    h.initiator,
    h.user_login,
    h.status,
    h.created_datetime,
    h.finished_datetime,
    h.start_date,
    h.end_date,
    h.remove_date
FROM ksk_report_header h
JOIN ksk_report_orchestrator o ON o.id = h.orchestrator_id
ORDER BY h.created_datetime DESC
LIMIT 50;
```

Фильтрация отчетов

```
-- Отчеты за сегодня
SELECT * FROM ksk_report_header
WHERE created_datetime >= CURRENT_DATE
ORDER BY created_datetime DESC;

-- Отчеты конкретного пользователя
SELECT * FROM ksk_report_header
WHERE user_login = 'ivanov'
ORDER BY created_datetime DESC;

-- Отчеты с ошибками
SELECT * FROM ksk_report_header
WHERE status = 'error'
ORDER BY created_datetime DESC;

-- Отчеты готовые к удалению
SELECT * FROM ksk_report_header
WHERE remove_date < CURRENT_DATE
ORDER BY remove_date;
```

Просмотр данных конкретного отчета

```
-- Определить тип отчета
SELECT
    h.id,
    o.report_code,
    o.report_table,
    h.name
```

```

FROM ksk_report_header h
JOIN ksk_report_orchestrator o ON o.id = h.orchestrator_id
WHERE h.id = 42;

-- Данные отчета (в зависимости от типа)
SELECT * FROM ksk_report_totals_data WHERE report_header_id = 42;
SELECT * FROM ksk_report_list_totals_data WHERE report_header_id = 42;
SELECT * FROM ksk_report_figurants_data WHERE report_header_id = 42;

```

Автоматизация

Ежедневные отчеты через cron

Bash-скрипт: /opt/ksk/scripts/daily_reports.sh

```

#!/bin/bash
# Ежедневная генерация отчетов КСК

PGHOST="localhost"
PGPORT="5432"
PGDATABASE="ksk_db"
PGUSER="ksk_user"

YESTERDAY=$(date -d "yesterday" +%Y-%m-%d)

echo "Генерация отчетов за ${YESTERDAY}..."

# 1. Общая статистика
psql -h $PGHOST -p $PGPORT -U $PGUSER -d $PGDATABASE -c \
"SELECT ksk_run_report('totals', 'system', NULL, '${YESTERDAY}', NULL, NULL);"

# 2. Итоги по спискам
psql -h $PGHOST -p $PGPORT -U $PGUSER -d $PGDATABASE -c \
"SELECT ksk_run_report('list_totals', 'system', NULL, '${YESTERDAY}', NULL, NULL);"

# 3. Статистика по типам платежей
psql -h $PGHOST -p $PGPORT -U $PGUSER -d $PGDATABASE -c \
"SELECT ksk_run_report('totals_by_payment_type', 'system', NULL, '${YESTERDAY}', NULL, NULL);"

echo "Отчеты созданы: $(date)"

```

Добавление в crontab:

```

# Ежедневно в 03:00
0 3 * * * /opt/ksk/scripts/daily_reports.sh >> /var/log/ksk_reports.log 2>&1

```

Еженедельные отчеты

Скрипт: /opt/ksk/scripts/weekly_reports.sh

```
#!/bin/bash

PGHOST="localhost"
PGPORT="5432"
PGDATABASE="ksk_db"
PGUSER="ksk_user"

START_DATE=$(date -d "last monday" +%Y-%m-%d)
END_DATE=$(date -d "last sunday" +%Y-%m-%d)

echo "Генерация недельных отчетов: ${START_DATE} - ${END_DATE}..."

# Отчет по фигурантам за неделю
psql -h $PGHOST -p $PGPORT -U $PGUSER -d $PGDATABASE -c \
"SELECT ksk_run_report('figurants', 'system', NULL, '${START_DATE}', '${END_DATE}', NULL)"

# Комбинированный отчет по спискам и типам
psql -h $PGHOST -p $PGPORT -U $PGUSER -d $PGDATABASE -c \
"SELECT ksk_run_report('list_totals_by_payment_type', 'system', NULL, '${START_DATE}', NULL)"

echo "Недельные отчеты созданы: $(date)"
```

Crontab:

```
# Каждый понедельник в 04:00
0 4 * * 1 /opt/ksk/scripts/weekly_reports.sh >> /var/log/ksk_reports.log 2>&1
```

SQL-скрипт для автоматизации

Файл: /opt/ksk/sql/generate_daily_reports.sql

```
-- Генерация ежедневных отчетов
DO $$

DECLARE
    v_date DATE := CURRENT_DATE - 1;
    v_report_id INTEGER;

BEGIN
    -- 1. Общая статистика
    SELECT ksk_run_report('totals', 'system', NULL, v_date, NULL, NULL)
    INTO v_report_id;
    RAISE NOTICE 'Создан отчет totals, ID: %', v_report_id;

    -- 2. Итоги по спискам
    SELECT ksk_run_report('list_totals', 'system', NULL, v_date, NULL, NULL)
    INTO v_report_id;
    RAISE NOTICE 'Создан отчет list_totals, ID: %', v_report_id;
```

```
-- 3. Статистика по типам платежей
SELECT ksk_run_report('totals_by_payment_type', 'system', NULL, v_date, NULL, NULL)
INTO v_report_id;
RAISE NOTICE 'Создан отчет totals_by_payment_type, ID: %', v_report_id;

RAISE NOTICE 'Все отчеты созданы успешно';
END $$;
```

Управление жизненным циклом

Очистка устаревших отчетов

Автоматическая очистка:

```
-- Удалить все отчеты с истекшим TTL
SELECT ksk_cleanup_old_reports();

-- Результат: количество удаленных отчетов
```

Скрипт очистки: /opt/ksk/scripts/cleanup_reports.sh

```
#!/bin/bash

PGHOST="localhost"
PGPORT="5432"
PGDATABASE="ksk_db"
PGUSER="ksk_user"

echo "Очистка устаревших отчетов..."

psql -h $PGHOST -p $PGPORT -U $PGUSER -d $PGDATABASE -c \
"SELECT ksk_cleanup_old_reports();"

echo "Очистка завершена: $(date)"
```

Crontab:

```
# Ежедневно в 02:00
0 2 * * * /opt/ksk/scripts/cleanup_reports.sh >> /var/log/ksk_cleanup.log 2>&1
```

Ручное удаление отчета

```
-- Удалить конкретный отчет
DELETE FROM ksk_report_header WHERE id = 42;

-- Данные отчета удаляются автоматически (CASCADE)
```

Изменение TTL для типа отчета

```
-- Изменить TTL для системных отчетов 'figurants'  
UPDATE ksk_report_orchestrator  
SET system_ttl = 90,  
    user_ttl = 14  
WHERE report_code = 'figurants';
```

Мониторинг

Статистика по отчетам

```
-- Количество отчетов по типам  
SELECT  
    o.name,  
    COUNT(*) AS total_reports,  
    COUNT(*) FILTER (WHERE h.status = 'done') AS success,  
    COUNT(*) FILTER (WHERE h.status = 'error') AS errors,  
    COUNT(*) FILTER (WHERE h.initiator = 'system') AS system_reports,  
    COUNT(*) FILTER (WHERE h.initiator = 'user') AS user_reports  
FROM ksk_report_header h  
JOIN ksk_report_orchestrator o ON o.id = h.orchestrator_id  
WHERE h.created_datetime >= CURRENT_DATE - 7  
GROUP BY o.name  
ORDER BY total_reports DESC;
```

Просмотр логов

```
-- Последние операции с отчетами  
SELECT  
    operation_name,  
    begin_time,  
    end_time,  
    duration,  
    status,  
    info  
FROM ksk_system_operations_log  
WHERE operation_code LIKE 'run_report%'  
ORDER BY begin_time DESC  
LIMIT 20;
```

Проверка размера данных отчетов

```
-- Размер таблиц с данными отчетов  
SELECT  
    schemaname,  
    tablename,  
    pg_size.pretty(pg_total_relation_size(schemaname || '.' || tablename)) AS size,
```

```
n_live_tup AS rows
FROM pg_stat_user_tables
WHERE tablename LIKE 'ksk_report_%_data'
ORDER BY pg_total_relation_size(schemaname||'.'||tablename) DESC;
```

Troubleshooting

Проблема: Отчет завис в статусе 'in_progress'

Причина: Ошибка выполнения функции генерации

Решение:

```
-- Найти зависшие отчеты
SELECT
    id,
    name,
    created_datetime,
    status
FROM ksk_report_header
WHERE status = 'in_progress'
    AND created_datetime < NOW() - INTERVAL '1 hour';

-- Проверить лог ошибок
SELECT
    operation_name,
    begin_time,
    status,
    err_msg
FROM ksk_system_operations_log
WHERE operation_code LIKE 'run_report%'
    AND status = 'error'
ORDER BY begin_time DESC
LIMIT 10;

-- Пометить как ошибку
UPDATE ksk_report_header
SET status = 'error',
    finished_datetime = NOW()
WHERE id = 42;
```

Проблема: Медленная генерация отчетов

Диагностика:

```
-- Проверить время выполнения
SELECT
    operation_name,
    begin_time,
    duration,
```

```

    info
FROM ksk_system_operations_log
WHERE operation_code LIKE 'run_report%'
ORDER BY duration DESC
LIMIT 10;

-- Проверить статистику таблиц
SELECT
    schemaname,
    tablename,
    last_analyze,
    n_live_tup
FROM pg_stat_user_tables
WHERE tablename IN ('ksk_result', 'ksk_figurant', 'ksk_figurant_match');

```

Решение:

```

-- Обновить статистику
ANALYZE ksk_result;
ANALYZE ksk_figurant;
ANALYZE ksk_figurant_match;

-- Проверить индексы
SELECT
    schemaname,
    tablename,
    indexname,
    idx_scan
FROM pg_stat_user_indexes
WHERE tablename IN ('ksk_result', 'ksk_figurant')
ORDER BY idx_scan;

```

Проблема: Не хватает места для отчетов

Диагностика:

```

-- Проверить количество отчетов
SELECT
    o.name,
    COUNT(*) AS reports,
    MIN(h.created_datetime) AS oldest,
    MAX(h.created_datetime) AS newest
FROM ksk_report_header h
JOIN ksk_report_orchestrator o ON o.id = h.orchestrator_id
GROUP BY o.name;

```

Решение:

```

-- Принудительная очистка старых отчетов
DELETE FROM ksk_report_header
WHERE created_datetime < CURRENT_DATE - 90;

```

```
-- Уменьшить TTL для пользовательских отчетов
UPDATE ksk_report_orchestrator
SET user_ttl = 3
WHERE report_code IN ('figurants', 'list_totals');
```

Рекомендуемое расписание операций

Операция	Периодичность	Время	Скрипт
Генерация ежедневных отчетов	Ежедневно	03:00	daily_reports.sh
Генерация недельных отчетов	Понедельник	04:00	weekly_reports.sh
Очистка устаревших отчетов	Ежедневно	02:00	cleanup_reports.sh
ANALYZE таблиц	Ежедневно	05:00	analyze_tables.sh

Дополнительные ресурсы

Полезные запросы

Экспорт отчета в CSV:

```
COPY (
    SELECT * FROM ksk_report_totals_data
    WHERE report_header_id = 42
) TO '/tmp/report_42.csv' CSV HEADER;
```

Сравнение отчетов за два дня:

```
WITH day1 AS (
    SELECT * FROM ksk_report_totals_data
    WHERE report_header_id = 41
),
day2 AS (
    SELECT * FROM ksk_report_totals_data
    WHERE report_header_id = 42
)
SELECT
    day2.total - day1.total AS delta_total,
    day2.total_review - day1.total_review AS delta_review,
    day2.total_deny - day1.total_deny AS delta_deny
FROM day1, day2;
```

История изменений

- **2025-10-25** - Создание документации
- **2025-10-25** - Добавлены примеры автоматизации
- **2025-10-25** - Добавлен раздел Troubleshooting

Готово! Полная документация по использованию системы отчетов КСК.

**

1. <https://learn.microsoft.com/ru-ru/sql/relational-databases/data-collection/system-data-collection-set-reports?view=sql-server-ver17>
2. <https://myseldon.com/ru/news/index/245467417>
3. <https://learn.microsoft.com/ru-ru/sql/reporting-services/report-design/expression-examples-report-builder-and-ssrs?view=sql-server-ver17>
4. <https://reg.cloud/support/cloud/oblastnyye-servery/ustanovka-programmnogo-obespecheniya/sql-vyrazheniya-primery-funkcij-bazy-dannyh>
5. <https://ivan-shamaev.ru/t-sql-fundamentals-and-examples/>
6. http://myvisualdatabase.com/doc_ru/button_action_report_sql.html
7. https://docs.tantorlabs.ru/tdb/ru/16_10/se1c/xfunc-sql.html
8. https://help.salesforce.com/s/articleView?id=sf.c360_a_create_a_calculated_insights_sql_function.htm&language=ru&type=5
9. <https://www.youtube.com/watch?v=rRCCvoSFTD8>
10. https://platformmv.sbertech.ru/docs/public/PSQ/6.4.3/common/documents/extensions/pg_walinspect.htm