Регулярные задачи обслуживания системы КСК

> **Версия:** 1.0
> **Дата:** 28 октября 2025
> **Проект:** Система контроля санкционных операций (КСК)
> **База данных:** PostgreSQL 13+

Автоматизированное обслуживание базы данных PostgreSQL для обеспечения стабильной работы системы КСК.

Сводная таблица расписания

Ежедневные задачи

#	Время	Задача	Функция/SQL	Приоритет	Время выполнения
1	00:30	ANALYZE вчерашних партиций	SQL-блок	Критический	~1 мин
2	01:00	Создание будущих партиций (7 дней)	`ksk_create_partitions_for_all_tables(CURRENT_DATE, 7)`	Высокий	~2 мин
3	01:30	Генерация системных отчётов	SQL-блок (цикл по оркестратору)	Высокий	~30 мин
4	02:00	Удаление прошлогодних партиций (>365 дней)	`ksk_drop_old_partitions(365)`	Средний	~5 мин
5	03:00	Удаление empty записей (>14 дней)	`ksk_cleanup_empty_records(14)`	Средний	~3 мин
6	03:30	Удаление пустых партиций (>14 дней)	`ksk_cleanup_empty_partitions('all', 14)`	Средний	~5 мин
7	04:00	Очистка старых отчётов (по TTL)	`ksk_cleanup_old_reports()`	Средний	~2 мин
8	04:30	Очистка системных логов (>365 дней)	`ksk_cleanup_old_logs(365)`	Низкий	~1 мин

#	Время	Задача	Функция/SQL	Приоритет	Время выполнения
9	05:00	VACUUM главных таблиц	SQL-блок (VACUUM ANALYZE)	Высокий	~5-10 мин

Итого: ~54-59 минут ежедневно

Еженедельные задачи

#	День	Время	Задача	Функция	Приоритет	Время выполнения
10	Воскресенье	04:00	Мониторинг bloat	`ksk_monitor_table_bloat()`	Средний	~30 сек

Детальное описание задач

Задача #1: ANALYZE вчерашних партиций

Время: 00:30
Приоритет: Критический
Назначение: Обновление статистики для планировщика запросов PostgreSQL

Описание:
Обновляет статистику распределения данных во вчерашних партициях. Критически важно для производительности утренних отчётов — без актуальной статистики планировщик может выбрать неоптимальные планы выполнения запросов.

SQL скрипт:

```
DO $$
DECLARE
  v_date TEXT := TO_CHAR(CURRENT_DATE - 1, 'YYYYMMDD');
BEGIN
  EXECUTE 'ANALYZE upoa_ksk_reports.part_ksk_result_' || v_date;
  EXECUTE 'ANALYZE upoa_ksk_reports.part_ksk_figurant_' || v_date;
  EXECUTE 'ANALYZE upoa_ksk_reports.part_ksk_match_' || v_date;
END $$;
```

Затрагиваемые объекты:

- `part_ksk_result_YYYYMMDD` — вчерашняя партиция результатов
- `part_ksk_figurant_YYYYMMDD` — вчерашняя партиция фигурантов
- `part_ksk_match_YYYYMMDD` — вчерашняя партиция совпадений

Эффект:

- Ускорение выполнения отчётов в 2-5 раз
- Корректный выбор индексов планировщиком
- Оптимизация JOIN-операций

Задача #2: Создание будущих партиций

Время: 01:00
Приоритет: Высокий
Назначение: Предварительное создание партиций на неделю вперёд

Описание:
Создаёт партиции на 7 дней вперёд для всех партиционированных таблиц. Предотвращает ошибки при вставке данных в будущие даты и снижает нагрузку в рабочее время.

SQL скрипт:

```
SELECT upoa_ksk_reports.ksk_create_partitions_for_all_tables(
  CURRENT_DATE,  -- базовая дата
  7              -- дней вперёд
);
```

Затрагиваемые таблицы:

- `ksk_result` — результаты проверок
- `ksk_figurant` — данные о фигурантах
- `ksk_match` — совпадения

Задача #3: Генерация системных отчётов

Время: 01:30
Приоритет: Высокий
Назначение: Автоматическая подготовка отчётов для утреннего использования

Описание:
Перебирает все `report_code` из таблицы `ksk_report_orchestrator` и генерирует системные отчёты. Отчёты создаются с `initiator = 'system'` и автоматически удаляются через 365 дней.

SQL скрипт:

```
DO $$
DECLARE
  rec RECORD;
BEGIN
  FOR rec IN
    SELECT report_code
    FROM upoa_ksk_reports.ksk_report_orchestrator
    ORDER BY report_code
  LOOP
    PERFORM upoa_ksk_reports.ksk_run_report(
      rec.report_code,
      'system'
    );
  END LOOP;
END $$;
```

Генерируемые отчёты:

- `totals` — общие итоги (заблокировано, пропущено, итого)
- `list_totals` — итоги по спискам санкций

- `totals_by_payment_type` — итоги по типам платежей
- `list_totals_by_payment_type` — детализация по спискам и типам

Задача #4: Удаление прошлогодних партиций

Время: 02:00
Приоритет: Средний
Назначение: Освобождение дискового пространства

Описание:
Удаляет партиции старше 365 дней для всех партиционированных таблиц.

SQL скрипт:

```
SELECT upoa_ksk_reports.ksk_drop_old_partitions(365);
```

Оценка освобождаемого места: ~50-100 GB в год

Задача #5: Удаление empty записей

Время: 03:00
Приоритет: Средний
Назначение: Очистка служебных записей

Описание:
Удаляет записи со статусом `resolution = 'empty'` старше 14 дней из таблицы `ksk_result`.

SQL скрипт:

```
SELECT upoa_ksk_reports.ksk_cleanup_empty_records(14);
```

Задача #6: Удаление пустых партиций

Время: 03:30
Приоритет: Средний
Назначение: Удаление партиций без данных

Описание:
Сканирует партиции старше 14 дней и удаляет те, которые не содержат данных.

SQL скрипт:

```
SELECT upoa_ksk_reports.ksk_cleanup_empty_partitions('all', 14);
```

Задача #7: Очистка старых отчётов

Время: 04:00
Приоритет: Средний
Назначение: Удаление отчётов по TTL

Описание:
Удаляет отчёты на основе времени жизни:

- Системные отчёты: 365 дней
- Пользовательские отчёты: 7-14 дней

SQL скрипт:

```
SELECT upoa_ksk_reports.ksk_cleanup_old_reports();
```

Задача #8: Очистка системных логов

Время: 04:30
Приоритет: Низкий
Назначение: Удаление старых записей логов

Описание:
Удаляет записи системного лога КСК старше 365 дней.

SQL скрипт:

```
SELECT upoa_ksk_reports.ksk_cleanup_old_logs(365);
```

Задача #9: VACUUM главных таблиц

Время: 05:00
Приоритет: Высокий
Назначение: Освобождение места и обновление статистики

Описание:
Выполняет VACUUM ANALYZE на главных таблицах после всех ночных cleanup-задач. Освобождает место от удалённых строк, обновляет visibility map и статистику планировщика.

SQL скрипт:

```
DO $$
BEGIN
  -- Партиционированные таблицы (включая все партиции автоматически)
  VACUUM ANALYZE upoa_ksk_reports.ksk_result;
  VACUUM ANALYZE upoa_ksk_reports.ksk_figurant;
  VACUUM ANALYZE upoa_ksk_reports.ksk_match;

  -- Таблицы отчётов
  VACUUM ANALYZE upoa_ksk_reports.ksk_report_header;
  VACUUM ANALYZE upoa_ksk_reports.ksk_system_operations_log;
END $$;
```

Важно: `VACUUM ANALYZE` на родительской партиционированной таблице автоматически обрабатывает все дочерние партиции.

🗓️ Еженедельные задачи

Задача #10: Мониторинг bloat

День: Воскресенье
Время: 04:00
Приоритет: Средний
Назначение: Контроль раздутия таблиц

Описание:
Проверяет процент "мёртвых" строк (bloat) во всех таблицах схемы `upoa_ksk_reports`. Логирует результаты в `ksk_system_operations_log`.

SQL скрипт:

```
SELECT upoa_ksk_reports.ksk_monitor_table_bloat();
```

Пороги:

- Здоровые таблицы: bloat <15% → статус `success`
- Предупреждение: bloat 15-30% → статус `success` с предупреждением
- Критично: bloat >30% → статус `error`

Просмотр результатов:

```
SELECT
  begin_time,
  status,
  info,
  errmsg
FROM upoa_ksk_reports.ksk_system_operations_log
WHERE operation_name LIKE '%bloat%'
ORDER BY begin_time DESC
LIMIT 10;
```

⚙️ Настройка автоматизации через cron

Редактирование crontab

```
crontab -e
```

Добавление задач

```
# Задача #1: ANALYZE вчерашних партиций (00:30)
30 0 * * * psql -d ksk_database -U postgres -c "DO \$\$ DECLARE v_date TEXT := TO_CHAR(CURRENT_DATE -

# Задача #2: Создание будущих партиций (01:00)
0 1 * * * psql -d ksk_database -U postgres -c "SELECT upoa_ksk_reports.ksk_create_partitions_for_all_
```

```
# Задача #3: Генерация системных отчётов (01:30)
30 1 * * * psql -d ksk_database -U postgres -c "DO \$\$ DECLARE rec RECORD; BEGIN FOR rec IN SELECT r

# Задача #4: Удаление прошлогодних партиций (02:00)
0 2 * * * psql -d ksk_database -U postgres -c "SELECT upoa_ksk_reports.ksk_drop_old_partitions(365);"

# Задача #5: Удаление empty записей (03:00)
0 3 * * * psql -d ksk_database -U postgres -c "SELECT upoa_ksk_reports.ksk_cleanup_empty_records(14);

# Задача #6: Удаление пустых партиций (03:30)
30 3 * * * psql -d ksk_database -U postgres -c "SELECT upoa_ksk_reports.ksk_cleanup_empty_partitions(

# Задача #7: Очистка старых отчётов (04:00)
0 4 * * * psql -d ksk_database -U postgres -c "SELECT upoa_ksk_reports.ksk_cleanup_old_reports();"

# Задача #8: Очистка системных логов (04:30)
30 4 * * * psql -d ksk_database -U postgres -c "SELECT upoa_ksk_reports.ksk_cleanup_old_logs(365);"

# Задача #9: VACUUM главных таблиц (05:00)
0 5 * * * psql -d ksk_database -U postgres -c "VACUUM ANALYZE upoa_ksk_reports.ksk_result; VACUUM ANA

# Задача #10: Мониторинг bloat (воскресенье 04:00)
0 4 * * 0 psql -d ksk_database -U postgres -c "SELECT upoa_ksk_reports.ksk_monitor_table_bloat();"
```

🔍 Мониторинг и проверка

Просмотр последних операций

```
SELECT
    begin_time,
    operation_name,
    status,
    info
FROM upoa_ksk_reports.ksk_system_operations_log
ORDER BY begin_time DESC
LIMIT 20;
```

Проверка ошибок за последние 24 часа

```
SELECT
    begin_time,
    operation_name,
    status,
    info,
    errmsg
FROM upoa_ksk_reports.ksk_system_operations_log
WHERE status = 'error'
  AND begin_time > NOW() - INTERVAL '24 hours'
ORDER BY begin_time DESC;
```

Статистика выполнения задач за месяц

```
SELECT
    DATE(begin_time) AS log_date,
    COUNT(*) AS operations_count,
    SUM(CASE WHEN status = 'success' THEN 1 ELSE 0 END) AS success_count,
    SUM(CASE WHEN status = 'error' THEN 1 ELSE 0 END) AS error_count
FROM upoa_ksk_reports.ksk_system_operations_log
WHERE begin_time > NOW() - INTERVAL '30 days'
GROUP BY DATE(begin_time)
ORDER BY log_date DESC;
```

⚠ Troubleshooting

Задача не выполнилась

1. Проверьте логи PostgreSQL: `/var/log/postgresql/postgresql-*.log`
2. Проверьте системный лог КСК:

   ```
   SELECT * FROM upoa_ksk_reports.ksk_system_operations_log
   WHERE status = 'error'
   ORDER BY begin_time DESC LIMIT 10;
   ```

3. Проверьте cron логи: `grep CRON /var/log/syslog`

Партиции не создаются

```
-- Проверка существующих партиций
SELECT * FROM upoa_ksk_reports.ksk_list_partitions('ksk_result');

-- Ручное создание
SELECT upoa_ksk_reports.ksk_create_partitions('ksk_result', CURRENT_DATE, 7);
```

VACUUM работает слишком долго

```
-- Проверка bloat
SELECT
    schemaname,
    tablename,
    n_dead_tup,
    n_live_tup,
    ROUND(100.0 * n_dead_tup / NULLIF(n_live_tup + n_dead_tup, 0), 2) AS dead_pct
FROM pg_stat_user_tables
WHERE schemaname = 'upoa_ksk_reports'
ORDER BY dead_pct DESC;
```

Отчёты не генерируются

```
-- Проверка оркестратора
SELECT * FROM upoa_ksk_reports.ksk_report_orchestrator;

-- Проверка последних отчётов
```

```
SELECT * FROM upoa_ksk_reports.ksk_report_header
ORDER BY create_date DESC LIMIT 10;
```

Контакты и поддержка

Команда: КСК Database Team
Email: ksk-db-support@example.com
Документация: Wiki проекта

Лицензия

⁂

1. https://www.postgresql.org/docs/current/maintenance.html
2. https://opensource-db.com/essential-postgresql-maintenance-activities-for-optimal-performance/
3. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/PostgreSQL_pg_cron.html
4. https://www.reddit.com/r/PostgreSQL/comments/x1jb6o/what_scheduled_maintenance_tasks_do_you_perform/
5. https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-maintenance
6. http://postgres-x2.github.io/reference/1.2/html/maintenance.html
7. https://www.redwood.com/article/job-scheduling-with-postgres/
8. https://visuresolutions.zendesk.com/hc/en-us/articles/4405371959187-Maintenance-of-PostgreSQL-Database-and-Automation
9. https://stackoverflow.com/questions/37122803/manual-schedule-in-postgresql