

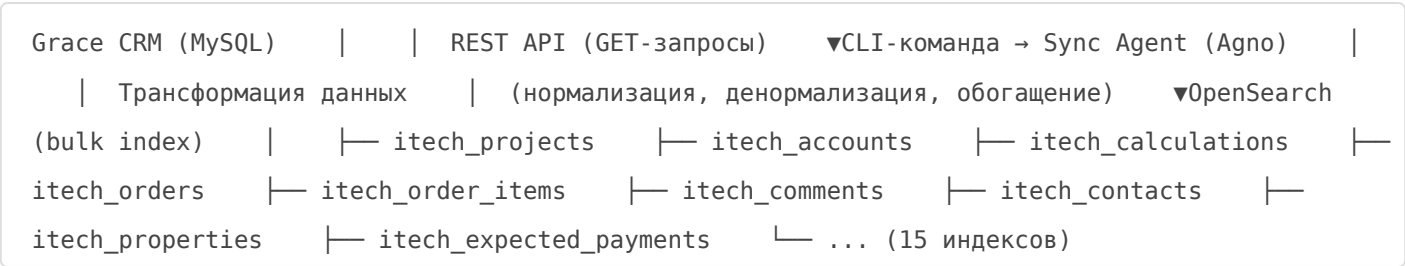
Схема интеграции Grace CRM → OpenSearch

Версия: 1.0 **Дата:** 4 мая 2026

Статус: Этап 0 -- согласование файл 02_integration_[schema.md](#)

1. Архитектура интеграции

Grace CRM (MySQL) не подключается к OpenSearch напрямую. Синхронизация осуществляется через Sync Agent -- Agno-агент, запускаемый по расписанию или вручную.



Важно: прямого доступа к MySQL Grace CRM у Sync Agent нет. Все данные получаются только через REST API Grace CRM.

2. Механизм запуска

2.1 CLI-команда

```
# Инкрементальная синхронизация (основной режим)
grace-sync run --mode incremental
# Полная синхронизация (ночная / восстановление)
grace-sync run --mode full
# Синхронизация конкретной сущности
grace-sync run --entity projects --mode incremental
# Ручной запуск с указанием временного окна
grace-sync run --mode incremental --since 2026-05-01T00:00:00
```

2.2 Расписание (cron)

Режим	Расписание	Описание
Инкрементальный	Каждые 15 минут	Основной рабочий режим
Полный	Ежедневно в 02:00	Восстановление консистентности
Ручной	По требованию администратора	После миграций, исправлений

3. Сущности и endpoints API

Сущность	Endpoint	Индекс OpenSearch	Частота
Сделки	GET /api/projects?updated_since=	itech_projects	15 мин
Контрагенты	GET /api/clients?updated_since=	itech_accounts	15 мин
Расчёты (КП)	GET /api/calculations?updated_since=	itech_calculations	15 мин
Заказы	GET /api/orders?updated_since=	itech_orders	15 мин
Позиции заказов	GET /api/order_items?updated_since=	itech_order_items	15 мин
Активности	GET /api/activities?updated_since=	itech_comments	15 мин
Задачи	GET /api/tasks?updated_since=	(в itech_projects)	15 мин
Контактные лица	GET /api/contacts?updated_since=	itech_contacts	1 час
Объекты	GET /api/objects?updated_since=	itech_properties	1 час
Платежи	GET /api/expected_payments?updated_since=	itech_expected_payments	1 час
Пользователи	GET /api/users	itech_users	1 раз/день

4. Режимы синхронизации

4.1 Инкрементальный режим

Принцип: запрашиваются только записи, изменённые с момента последней синхронизации.

```
1. Читаем last_sync_timestamp из хранилища состояния
2. GET /api/{entity}?updated_since={last_sync_timestamp}
3. Трансформируем полученные записи
4. Bulk upsert в OpenSearch (upsert по id)
5. Обновляем last_sync_timestamp = текущее время
```

Хранение состояния: файл `sync_state.json` или переменная окружения. Формат:

```
{ "projects": "2026-05-04T08:45:00", "clients": "2026-05-04T08:45:00", "orders": "2026-05-04T08:45:00" }
```

4.2 Полный режим

Принцип: полная выгрузка всех сущностей с пагинацией, полная пересборка индексов.

```
1. Создаём новый индекс с суффиксом _tmp
2. Загружаем все данные через API с пагинацией (batch по 500 записей)
3. После успешной загрузки — переключаем алиас
4. Удаляем старый индекс
```

Применяется при: первоначальном развёртывании, восстановлении после сбоя, изменении маппинга.

5. Трансформация данных

5.1 Денормализация

При индексации данные обогащаются связанными сущностями во избежание JOIN-запросов в OpenSearch:

```
projects → добавляем account_name, manager_name, property_name, project_status
(название)
orders → добавляем account_name, manager_name, company_name
comments → добавляем
```

project_name, project_status, account_name, manager_name

5.2 Вычисляемые поля

Поле	Индекс	Формула
is_shipped	itech_orders, itech_order_items	shipping_date_fact IS NOT NULL
revenue_mln	itech_accounts	SUM(order_items.amount) / 1 000 000
debt_mln	itech_accounts	SUM(expected_payments где paid_amount < pay_amount) / 1 000 000
order_count	itech_accounts	COUNT(orders.id)
top_account	itech_properties	Контрагент с наибольшим числом проектов на объекте

5.3 Нормализация данных

- Имена менеджеров: нормализация пробелов, удаление неразрывных пробелов (\xa0)
- Даты: приведение к формату yyyy-MM-dd HH:mm:ss
- Суммы: NULL -> 0.0
- Булевы поля: 0/1 -> false/true

6. Обработка конфликтов и ошибок

6.1 Стратегия при конфликте версий

Правило: последнее изменение по `updated_at` побеждает.

```
# При upsert в OpenSearch{ "doc": { ...новые поля... }, "doc_as_upsert": True}
```

Если `updated_at` в новой записи меньше, чем в существующей -- запись не обновляется (idempotent операция).

6.2 Обработка ошибок API

Ситуация	Поведение
HTTP 429 (rate limit)	Пауза 60 сек, повтор
HTTP 5xx (ошибка сервера)	Retry 3 раза с экспоненциальной задержкой
HTTP 404 (запись удалена)	Удаление из OpenSearch
Таймаут соединения	Retry 3 раза, затем запись в лог ошибок
Частичный сбой батча	Повтор только ошибочных записей

6.3 Логирование

Каждый запуск синхронизации фиксирует:

```
{ "run_id": "uuid", "mode": "incremental", "started_at": "2026-05-04T09:00:00",
  "finished_at": "2026-05-04T09:00:23", "entities": {    "projects": { "fetch": 42,
"indexed": 42, "errors": 0 },    "orders": { "fetch": 7, "indexed": 7, "errors": 0 }  },
  "status": "success"}
```

7. Мониторинг и алерты

Метрика	Пороговое значение	Действие
Время последней синхронизации	> 30 минут назад	Алерт администратору
Количество ошибок за запуск	> 5% от batch	Алерт + запись в лог
Расхождение счётчиков	Индекс < 90% от MySQL	Запуск полной синхронизации
Недоступность API Grace	> 3 неудачных попытки	Алерт + пауза 10 минут

8. Требования к API Grace CRM

Для корректной работы интеграции API Grace CRM должен поддерживать:

Требование	Параметр	Обязательно
------------	----------	-------------

Фильтрация по дате изменения	<code>?updated_since=IS08601</code>	Да
Пагинация	<code>?page=N&per_page=500</code>	Да
Получение записи по ID	<code>GET /{entity}/{id}</code>	Да
Аутентификация	API-ключ в заголовке	Да
Rate limit	Информация о лимитах	Да (для настройки задержек)

Открытый вопрос: наличие webhooks в API Grace CRM уточняется на Этапе 0. При наличии webhooks -- добавить режим event-driven синхронизации в дополнение к cron-polling.

Revision #1
Created 2026-06-22 03:22:42 UTC by Claude Ops
Updated 2026-06-22 03:22:42 UTC by Claude Ops