

Business CRM

Техническая спецификация БД: Business CRM

Контекст: Business Admin — панель управления для B2B клиентов экосистемы Yalla.

0. Словарь Сущностей (Entity Reference)

Перед погружением в схему, зафиксируем назначение ключевых объектов системы.

Сущность	Описание	Роль в системе	Критичность
Company	Юридическое лицо (B2B-клиент). "Владелец" денег и сотрудников.	Ядро (Core)	 High
Company Address	Филиал/Офис компании. Точка доставки.	Логистика	 Med
B2B User	Пользователь админки (HR, Бухгалтер). Управляет кабинетом, но не ест.	Доступ (Access)	 Med
Audit Log	Журнал безопасности. "Кто изменил лимит директора?".	Безопасность	 Low
Employee	Конечный пользователь (Сотрудник). Тот, кто физически ест обеды и пользуется Client App.	Персонал (Staff)	 High
Lunch Subscription	"Намерение" сотрудника питаться. Шаблон для автоматической генерации заказов.	Операции	 Med
Daily Order	Фактический заказ еды на конкретную дату. Финансовое обязательство.	Операции	 High

Сущность	Описание	Роль в системе	Критичность
Invoice	Счёт на оплату от Yalla Lunch. Основание для пополнения баланса.	Финансы	🟡 Med
Company Transaction	Запись в главной книге (Ledger). Любое движение средств.	Финансы	🔥 High
Company Document	Файл (Акт сверки, Договор).	Документооборот	🟢 Low
System News	Новости сервиса для администраторов.	Инфо-центр	🟢 Low



1. ER-Диаграмма (High Level)

erDiagram

%% Core & Access

```
COMPANIES ||--|{ B2B_USERS : manages
COMPANIES ||--|{ COMPANY_ADDRESSES : has
COMPANIES ||--|{ COMPANY_DOCUMENTS : owns
```

%% Staff & Dining

```
COMPANIES ||--|{ EMPLOYEES : employs
EMPLOYEES ||--|| LUNCH_SUBSCRIPTIONS : has
EMPLOYEES ||--|{ DAILY_ORDERS : receives_benefit
```

%% Finance

```
COMPANIES ||--|{ INVOICES : billed_to
COMPANIES ||--|{ COMPANY_TRANSACTIONS : ledger_entry
```

%% Communication

```
SYSTEM_NEWS ||--|{ NEWS_READ_STATUS : read_by
B2B_USERS ||--|{ NEWS_READ_STATUS : reads
```

%% Relations Details

```
INVOICES ||--|{ COMPANY_TRANSACTIONS : pays
```

```
DAILY_ORDERS ||--|{ COMPANY_TRANSACTIONS : charges  
  
%% Guest Orders Link  
B2B_USERS ||--o{ DAILY_ORDERS : created_guest_order  
  
companies {  
    bigint id PK  
    decimal balance  
    decimal overdraft_limit  
    string timezone  
}
```

2. Детальная SQL Схема и Рекомендации

Модуль 1: Core (Компании и Филиалы)

companies

Юридическое лицо. "Кошелек" системы.

```
CREATE TABLE companies (  
    id BIGSERIAL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
  
    -- Статусы  
    status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE, BLOCKED_DEBT, ARCH  
    IVED  
  
    -- Финансы  
    balance DECIMAL(15, 2) DEFAULT 0.00, -- Текущий депозит (Может быть  
    < 0)  
    overdraft_limit DECIMAL(15, 2) DEFAULT 0.00, -- Разрешенный минус (нап  
    п. 5000)  
    currency_code CHAR(3) DEFAULT 'TJS', -- ISO 4217  
  
    -- Настройки Логистики
```

```
    timezone VARCHAR(50) DEFAULT 'Asia/Dushanbe', -- ⚠ Критично для Cut
    -off
    cutoff_time TIME DEFAULT '10:30', -- Время блокировки заказов на сегод
    ня

    -- Метаданные
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    deleted_at TIMESTAMP WITH TIME ZONE -- Soft Delete
);
```

💡 Комментарии архитектора:

- `balance`: Мы разрешаем балансу уходить в минус, но не ниже `overdraft_limit`. Это позволяет не блокировать питание сотрудников мгновенно, если бухгалтер забыл оплатить счет на 1 день ("Мягкая блокировка").
- `timezone`: У клиентов могут быть офисы в разных часовых поясах (или странах в будущем). Правило "Заказы до 10:30" должно считаться по **их** локальному времени, а не по серверному времени в data-центре.

⚙️ Рекомендации (Dev Ops & Indexing):

- **Index:** `CREATE INDEX idx_companies_status ON companies(status);` — для быстрой выборки активных клиентов кроном генерации заказов.
- **Constraint:** Рекомендую добавить `CHECK (balance >= -overdraft_limit)`. БД сама не даст списать деньги сверх лимита, даже если в коде ошибка.

company_addresses

Адреса доставки (Офисы, Склады).

```
CREATE TABLE company_addresses (
    id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL REFERENCES companies(id),

    name VARCHAR(100) NOT NULL, -- "Главный офис", "Цех №2"
    address_line TEXT NOT NULL, -- Точный адрес для курьера
```

```
geo_point POINT, -- (lat, long) для карты
```

```
is_default BOOLEAN DEFAULT FALSE, -- Подставляется при создании новых сотрудников
```

```
deleted_at TIMESTAMP WITH TIME ZONE, -- ⚠ Soft Delete обязателен  
created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
```

```
);
```

💡 Комментарии архитектора:

- `deleted_at`: Если офис переехал, мы не удаляем запись, а помечаем удаленной. Это нужно, чтобы в истории заказов за прошлый год сохранился старый адрес доставки. Физическое удаление сломает целостность данных (`daily_orders` ссылаются сюда).

⚙️ Рекомендации:

- **Querying:** Все SELECT запросы для UI должны содержать `WHERE deleted_at IS NULL`.
- **Geo:** Если планируем карту, стоит рассмотреть PostGIS расширение, но для MVP хватит типа `POINT`.

👥 Модуль 2: Access (Пользователи Админки)

b2b_users

Администраторы (HR, Бухгалтер).

```
CREATE TABLE b2b_users (  
    id BIGSERIAL PRIMARY KEY,  
    company_id BIGINT NOT NULL REFERENCES companies(id),
```

```
    full_name VARCHAR(150) NOT NULL,  
    phone VARCHAR(20) NOT NULL UNIQUE, -- Логин (Формат E.164)  
    email VARCHAR(150),  
    password_hash VARCHAR(255),
```

```
-- Ролевая модель
role VARCHAR(50) NOT NULL, -- 'SUPERADMIN', 'ADMIN', 'ACCOUNTANT',
'OFFICE_MANAGER'
permissions JSONB DEFAULT '{}', -- Гибкие права: {"can_manage_financ
e": true, "can_edit_staff": false}

status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE, BLOCKED
last_login_at TIMESTAMP WITH TIME ZONE,
created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```

Комментарии архитектора:

- `permissions` (JSONB): Позволяет создавать кастомные роли без миграций БД. Например, дать Бухгалтеру доступ к "Сотрудникам", но только read-only.
- **Связь с сотрудниками:** Этот пользователь никак не связан с таблицей `employees`. Админ может быть сотрудником (есть), а может быть внешним бухгалтером на аутсорсе (не ест).

Рекомендации:

- **Index:** `CREATE INDEX idx_b2b_users_company ON b2b_users(company_id);`
- **Security:** Пароли хранить только в хэшированном виде (Argon2 или Bcrypt).

audit_logs

Журнал действий.

```
CREATE TABLE audit_logs (
id BIGSERIAL PRIMARY KEY,
company_id BIGINT NOT NULL REFERENCES companies(id),
actor_id BIGINT REFERENCES b2b_users(id), -- Кто сделал
```

```
event_type VARCHAR(50) NOT NULL, -- CREATE, UPDATE, DELETE, FORCE  
_PAUSE  
entity_type VARCHAR(50) NOT NULL, -- EMPLOYEE, BUDGET, ORDER, INV  
OICE  
entity_id BIGINT NOT NULL,  
  
changes JSONB, -- Diff: {"field": "daily_limit", "old": 50, "new": 100}  
ip_address INET,  
  
created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

 Комментарии архитектора:

- Это "черный ящик" для разбора инцидентов. Записи отсюда удалять нельзя (Append Only).
- JSONB `changes` позволяет хранить структуру изменений любой сущности без создания отдельных таблиц истории.

 Рекомендации:

- **Partitioning:** Эта таблица будет расти очень быстро. Рекомендую сразу сделать **партиционирование по дате** (например, по месяцам). Иначе через год запросы к логам будут висеть и "съедать" память.

Модуль 3: Staff (Сотрудники)

employees

Конечные потребители.

```
CREATE TABLE employees (  
id BIGSERIAL PRIMARY KEY,  
company_id BIGINT NOT NULL REFERENCES companies(id),  
  
-- Профиль  
full_name VARCHAR(150) NOT NULL,
```

```
    phone VARCHAR(20) NOT NULL UNIQUE, --  ГЛАВНЫЙ КЛЮЧ СВЯЗИ С  
CLIENT APP  
    email VARCHAR(150),  
    position VARCHAR(100),  
  
    -- Статусы  
    status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE, DEACTIVATED (Увое  
н)  
  
    -- Инвайты (Процесс приглашения в Client App)  
    invite_status VARCHAR(50) DEFAULT 'PENDING', -- PENDING → SENT → A  
CCEPTED / REJECTED  
    invite_token VARCHAR(100), -- Токен для deep-link  
    invite_sent_at TIMESTAMP WITH TIME ZONE,  
    joined_at TIMESTAMP WITH TIME ZONE, -- Дата первого входа  
  
    -- Бюджетная политика (Настройки)  
    budget_total DECIMAL(10, 2) DEFAULT 0.00, -- Лимит на период  
    budget_period VARCHAR(20) DEFAULT 'MONTH', -- MONTH, WEEK, DAY  
    daily_limit DECIMAL(10, 2) DEFAULT 0.00, -- Лимит на день  
    auto_renew BOOLEAN DEFAULT TRUE, -- Обновлять ли лимит автоматич  
ески  
  
    -- Операционные данные (Состояние кошелька)  
    current_period_spend DECIMAL(10, 2) DEFAULT 0.00, -- Сколько УЖЕ пот  
рачено в этом периоде  
  
    version INT DEFAULT 1, -- Optimistic Lock  
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),  
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

Комментарии архитектора:

- **phone** (Unique): Гарантирует "Strict Uniqueness". Один человек не может работать в двух компаниях одновременно. Это упрощает логику Client

App (не нужно выбирать "От какой компании я сейчас ем?").

- `version` : Используем Optimistic Locking. При обновлении бюджета проверяем `WHERE version = X`. Если версия в БД уже изменилась (кто-то другой отредактировал в ту же секунду), транзакция откатывается.
- `current_period_spend` : Мы храним "траты", а не "остаток". Остаток вычисляется как `budget_total - current_period_spend`. Это безопаснее: если Админ увеличит лимит посреди месяца, остаток пересчитается корректно сам.

Рекомендации:

- **Index:** `CREATE INDEX idx_employees_phone ON employees(phone);` — для мгновенного поиска при логине в Client App.

Модуль 4: Operations (Заказы)

`lunch_subscriptions`

Шаблоны регулярного питания.

```
CREATE TABLE lunch_subscriptions (
    id BIGSERIAL PRIMARY KEY,
    employee_id BIGINT NOT NULL REFERENCES employees(id),
    company_address_id BIGINT NOT NULL REFERENCES company_addresses
        (id),
    combo_type VARCHAR(50) NOT NULL, -- 'COMBO_25', 'COMBO_35'
    status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE, PAUSED_PERMANENT

    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    CONSTRAINT uq_emp_subscription UNIQUE (employee_id)
);
```

Комментарии архитектора:

- Связь `One-to-One` с сотрудником. Мы не храним историю изменений подписки здесь (история будет в `audit_logs`).
- Это таблица "Намерений". Сами заказы генерируются в `daily_orders`.

daily_orders

Фактические заказы на день.

```
CREATE TABLE daily_orders (
    id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL REFERENCES companies(id),
    order_date DATE NOT NULL, -- На какую дату заказ

    -- Кто ест (Nullable для Гостей)
    employee_id BIGINT REFERENCES employees(id),
    is_guest BOOLEAN DEFAULT FALSE,
    guest_name VARCHAR(150), -- "Делегация из банка"
    created_by_user_id BIGINT REFERENCES b2b_users(id), -- Кто из админов
    создал гостевой заказ

    -- Детали
    combo_type VARCHAR(50) NOT NULL,
    company_address_id BIGINT REFERENCES company_addresses(id),

    -- Финансы (Снэпшот)
    price_at_moment DECIMAL(10, 2) NOT NULL, -- Фиксированная цена на м
    омент генерации
    currency_code CHAR(3) DEFAULT 'TJS',

    -- Статус
    status VARCHAR(50) DEFAULT 'NEW', -- NEW, PAUSED_ONCE (Пауза на се
    годня), DELIVERED, CANCELED

    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```



Комментарии архитектора:

- `price_at_moment` : Критически важно сохранять цену на момент создания заказа. Цены в меню могут меняться (инфляция), но уже созданный

заказ не должен менять стоимость задним числом.

- `is_guest`: Позволяет делать заказы "на компанию" (для партнеров, гостей), не привязывая их к конкретному сотруднику.

Поле `created_by_user_id` обязательно для аудита таких заказов.

Рекомендации:

- **Index:** `CREATE INDEX idx_daily_orders_lookup ON daily_orders(company_id, order_date);` — основной индекс для Админки ("Показать таблицу заказов на сегодня").

Модуль 5: Finance (Финансы и Документы)

invoices

Счета на оплату.

```
CREATE TABLE invoices (
    id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL REFERENCES companies(id),
    external_id VARCHAR(100) UNIQUE, -- ID из Yalla CRM / 1C

    amount DECIMAL(15, 2) NOT NULL,
    status VARCHAR(50) DEFAULT 'UNPAID', -- UNPAID, PAID, CANCELLED, OVERDUE

    due_date DATE,
    paid_at TIMESTAMP WITH TIME ZONE,

    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```

Комментарии архитектора:

- `external_id`: Ключ идемпотентности. Если Yalla CRM случайно пришлет один и тот же вебхук создания счета дважды (сетевой сбой), БД отклонит дубль благодаря UNIQUE constraint.

company_documents

Файловое хранилище.

```
CREATE TABLE company_documents (
    id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL REFERENCES companies(id),
    type VARCHAR(50) NOT NULL, -- 'ACT_OF_RECONCILIATION', 'INVOICE_PDF', 'CONTRACT'
    file_url TEXT NOT NULL, -- S3 URL
    period_start DATE, -- Для актов сверки
    period_end DATE,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```

Рекомендации:

- **Security:** `file_url` должен быть не публичным, а временным (Signed URL) или проксируться через бэкенд с проверкой прав доступа. Нельзя давать прямые ссылки.

company_transactions

Главная книга (Ledger).

```
CREATE TABLE company_transactions (
    id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL REFERENCES companies(id),
    type VARCHAR(50) NOT NULL,
    -- 'DEPOSIT' (Пополнение),
    -- 'LUNCH_DEDUCTION' (Списание за обед),
    -- 'GUEST_ORDER' (Списание за гостя),
    -- 'CLIENT_APP_ORDER' (Списание из приложения),
    -- 'REFUND' (Возврат)
```

```

amount DECIMAL(15, 2) NOT NULL, -- (-) Списание, (+) Пополнение

-- Связи (Traceability)
invoice_id BIGINT REFERENCES invoices(id),
daily_order_id BIGINT REFERENCES daily_orders(id),
client_app_order_uuid UUID, -- Ссылка на внешний заказ из Client App

balance_after DECIMAL(15, 2) NOT NULL, -- Snapshot баланса

created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

```

 Комментарии архитектора:

- Это "святая святых". Записи здесь **никогда** не изменяются (Immutable).
- `balance_after` : Позволяет быстро проверить целостность. `balance_after` текущей записи должен быть равен `balance_after` предыдущей + `amount`. Если не сходится — значит, кто-то лазил в базу руками (или баг).

Модуль 6: Info-Center (Новости)

`system_news`

Новости сервиса.

```

CREATE TABLE system_news (
    id BIGSERIAL PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    content TEXT NOT NULL,
    is_published BOOLEAN DEFAULT FALSE,
    published_at TIMESTAMP WITH TIME ZONE,

    target_roles VARCHAR(50)[], -- Массив ролей: ['ADMIN', 'ACCOUNTANT']
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

```

news_read_status

Трекинг прочтения.

```
CREATE TABLE news_read_status (
    news_id BIGINT REFERENCES system_news(id),
    user_id BIGINT REFERENCES b2b_users(id),
    read_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    PRIMARY KEY (news_id, user_id)
);
```

⚠ 3. Checklist для Разработчиков (Developer Rules)

🔴 Critical (Безопасность и Деньги)

1. **Strict Uniqueness:** Проверять уникальность телефона сотрудника перед вставкой. Если телефон занят — выдавать четкую ошибку "Сотрудник уже в другой компании".
2. **Cut-off Time:** Использовать `companies.timezone` для проверок времени, а не серверное время. 10:30 утра — это разные моменты в Душанбе и Лондоне.
3. **Transactions:** Любое изменение баланса (`companies.balance`) должно быть обернуто в транзакцию БД вместе с `INSERT` в `company_transactions`.

🟡 Warning (Логика)

1. **Soft Deletes:** Учитывать `deleted_at IS NULL` во всех выборках сущностей (Address, Company).
2. **Snapshotting:** При создании `daily_orders` копировать цену, адрес и имя сотрудника.
3. **Invites:** Сотрудник становится активным (`is_active = TRUE`) только после первого входа в Client App. До этого момента на него нельзя генерировать заказы (иначе еда приедет призраку).

🔵 Suggestions (Оптимизация)

- 1. Partitioning:** Рассмотреть
партиционирование `daily_orders` И `company_transactions` по годам для ускорения
выборки, когда база разрастется.
- 2. UUID for External Links:** Использовать UUID для поля `client_app_order_uuid`, так
как это слабая ссылка на другую БД.