










# Yalla CRM

## Техническая спецификация БД: Yalla Core CRM

**Контекст:** Центральная нервная система (ERP). Источник правды (MDM), центр логистики и финансов.

### 0. Словарь Сущностей (Entity Reference)

Сущность	Описание	Тип
<b>Partner</b>	Любой поставщик еды (Кухня Yalla, Ресторан, Ритейл). Заменяет старую сущность <b>Kitchen</b> .	 Master Data
<b>Company</b>	B2B Клиент. Заказчик обедов.	 Master Data
<b>System User</b>	Сотрудник Yalla (Админ, Диспетчер, Сейлз).	 Access
<b>Courier</b>	Сотрудник логистики.	 Logistics
<b>Daily Menu</b>	План производства еды на дату (для B2B).	 Legacy B2B
<b>Daily Company Order</b>	Сводный заказ компании на день.	 Legacy B2B
<b>Delivery Task</b>	Универсальная задача на доставку (Маршрутный лист или Разовый заказ).	 Logistics
<b>Global Product</b>	Эталонный товар (PIM). База знаний о еде.	 Catalog
<b>Core Transaction</b>	Финансовая проводка взаиморасчетов с Партнерами и Компаниями.	 Finance

### 1. ER-Диаграмма (Target Architecture)

erDiagram

%% CORE MDM

PARTNERS ||--|{ PARTNER\_LOCATIONS : has

COMPANIES ||--|{ COMPANY\_CONTRACTS : signs

SYSTEM\_USERS ||--|{ COMPANIES : manages\_sales

```

%% CATALOG PIM
GLOBAL_CATEGORIES ||--|{ GLOBAL_PRODUCTS : classifies
PARTNERS ||--|{ PARTNER_MENU_ITEMS : sells
GLOBAL_PRODUCTS ||--o{ PARTNER_MENU_ITEMS : linked_to

%% LOGISTICS (UNIFIED)
COURIERS ||--|{ SHIFTS : works
COURIERS ||--|{ DELIVERY_TASKS : assigned_to
DELIVERY_TASKS }|--|| ORDERS_B2C_LINK : fulfils
DELIVERY_TASKS }|--|| ORDERS_B2B_LINK : fulfils

%% LEGACY B2B ENGINE (REFACTORED)
COMPANIES ||--|{ DAILY_COMPANY_ORDERS : places
PARTNERS ||--|{ DAILY_MENUS : cooks
DAILY_MENUS ||--|{ DAILY_COMPANY_ORDERS : content
DAILY_COMPANY_ORDERS ||--|| DELIVERY_TASKS : generates

%% SUPPORT
SYSTEM_USERS ||--|{ INCIDENTS : resolves
PARTNERS ||--o{ INCIDENTS : target

partners {
    bigint id PK
    string type "INTERNAL/EXTERNAL"
    decimal balance
}

```

## 2. Детальная SQL Схема и Комментарии



### Модуль 1: MDM (Master Data Management)

Мы отказываемся от таблицы **ACTORS** для новых сущностей в пользу строгих типов. Но данные из старой БД должны быть мигрированы сюда.

**partners** (ex-Kitchens)

Единый реестр всех, кто готовит или продает. Миграция: `ACTORS (type='KITCHEN')` -  
> `partners (type='INTERNAL_KITCHEN')` .

```
CREATE TABLE partners (  
  id BIGSERIAL PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  
  -- Типология  
  type VARCHAR(50) NOT NULL, -- 'INTERNAL_KITCHEN' (Цех Yalla), 'REST  
  AURANT' (Партнер), 'RETAIL_STORE'  
  status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE, ONBOARDING, BLOCK  
  ED  
  
  -- Финансы (Взаиморасчеты)  
  legal_entity_name VARCHAR(255),  
  commission_rate DECIMAL(5, 2) DEFAULT 0.00, -- % Yalla  
  balance DECIMAL(15, 2) DEFAULT 0.00, -- Сколько Yalla должна партнеру  
  (или он нам)  
  
  -- Интеграция  
  merchant_admin_store_uuid UUID, -- Ссылка на Merchant CRM (для синхр  
  онизации меню/стопов)  
  
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),  
  updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

### `companies` (Master)

Мастер-таблица B2B клиентов. Синхронизация: Это "Мастер". При изменении  
здесь → шлем вебхук в `Business CRM` .

```
CREATE TABLE companies (  
  id BIGSERIAL PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,
```

```

-- Sales Funnel (Наследие LEADS)
sales_manager_id BIGINT REFERENCES system_users(id), -- Кто ведет клиента
lifecycle_stage VARCHAR(50) DEFAULT 'LEAD', -- LEAD, NEGOTIATION, ACTIVE_CLIENT, CHURNED

-- Настройки
contract_number VARCHAR(100),
payment_terms VARCHAR(50) DEFAULT 'PREPAID', -- PREPAID (Депозит), POSTPAID (Кредит)

-- Интеграция
business_crm_company_id BIGINT, -- ID зеркала в Business CRM

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

```

## Модуль 2: Catalog (PIM - Product Information)

Центральный каталог. Критичен для Ритейла, чтобы не забивать товары вручную в каждом магазине.

### **global\_categories**

Дерево категорий. Раздается во все клиентские приложения.

```

CREATE TABLE global_categories (
  id SERIAL PRIMARY KEY,
  name_ru VARCHAR(100) NOT NULL,
  name_tj VARCHAR(100),
  icon_url TEXT,
  parent_id INT REFERENCES global_categories(id), -- Иерархия
  sort_order INT DEFAULT 0
);

```

## global\_products

Эталонные товары ("База знаний"). *Логика:* Магазин сканирует штрихкод → Мы находим товар здесь → Создаем карточку в магазине.

```
CREATE TABLE global_products (  
  id BIGSERIAL PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  barcode VARCHAR(50) UNIQUE, -- EAN-13  
  
  category_id INT REFERENCES global_categories(id),  
  
  description TEXT,  
  image_url TEXT,  
  
  -- Характеристики  
  weight_g INT,  
  is_alcohol BOOLEAN DEFAULT FALSE,  
  
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

## Модуль 3: Logistics Core (Unified)

Мы заменяем старую сущность **DELIVERY\_ROUTES** на более гибкую систему задач, способную возить и "коробки с обедами", и "пиццу за 30 минут".

## couriers (ex-Actors)

Сотрудники доставки.

```
CREATE TABLE couriers (  
  id BIGSERIAL PRIMARY KEY,  
  full_name VARCHAR(150) NOT NULL,  
  phone VARCHAR(20) NOT NULL UNIQUE,  
  
  status VARCHAR(50) DEFAULT 'OFFLINE', -- OFFLINE, ONLINE, BUSY (Везе
```

```

т)
    current_geo_point POINT, -- Последняя известная локация (для диспетче
ра)

    transport_type VARCHAR(50) DEFAULT 'SCOOTER', -- FOOT, SCOOTER, CA
R

    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

```

### **delivery\_tasks** (Unified Task)

Единица работы курьера. *Полиморфизм*: Может быть связана с "Сводным заказом компании" (B2B) или "Частным заказом" (B2C).

```

CREATE TABLE delivery_tasks (
    id BIGSERIAL PRIMARY KEY,

    -- Исполнитель
    courier_id BIGINT REFERENCES couriers(id),

    -- Точка А (Забор)
    pickup_partner_id BIGINT NOT NULL REFERENCES partners(id), -- Откуда
забираем
    pickup_address_text TEXT NOT NULL,
    pickup_deadline TIMESTAMP WITH TIME ZONE, -- Когда забрать (SLA)

    -- Точка Б (Вручение)
    dropoff_address_text TEXT NOT NULL,
    dropoff_geo_point POINT, -- Для карты
    dropoff_deadline TIMESTAMP WITH TIME ZONE, -- Когда отдать

    -- Контекст (Что везем?)
    source_type VARCHAR(50) NOT NULL, -- 'B2B_COMPANY_ORDER', 'B2C_C
LIENT_ORDER'
    source_id VARCHAR(100) NOT NULL, -- ID из Legacy Order или Merchant

```

## Order

```
-- Статус
status VARCHAR(50) DEFAULT 'PENDING', -- PENDING → ASSIGNED → PICKED_UP → DELIVERED

created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
-- Индекс для Диспетчера: "Покажи все незакрытые доставки"
CREATE INDEX idx_delivery_tasks_active ON delivery_tasks(status) WHERE status NOT IN ('DELIVERED', 'FAILED');
```

## Модуль 4: Legacy B2B (Refactored)

Мы сохраняем бизнес-логику "Проектов" и "Меню по дням", но переводим связи на новые таблицы.

### daily\_menus

План производства на дату. Связь: Теперь ссылается на `partners` (где `type='INTERNAL_KITCHEN'`).

```
CREATE TABLE daily_menus (
  id BIGSERIAL PRIMARY KEY,
  menu_date DATE NOT NULL,


  partner_id BIGINT NOT NULL REFERENCES partners(id), -- Кто готовит
  global_product_id BIGINT REFERENCES global_products(id), -- Что готови
  М

  price DECIMAL(10, 2) NOT NULL,
  planned_quantity INT NOT NULL, -- План ("Сколько порций сделать")
  available_quantity INT NOT NULL, -- Факт ("Сколько осталось")
);
```

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

### daily\_company\_orders

Сводный заказ компании на день. *Логика:* В 10:30 утра (Cut-off) система берет все индивидуальные заказы из **Business CRM** и агрегирует их в одну запись здесь для кухни.

```
CREATE TABLE daily_company_orders (  
  id BIGSERIAL PRIMARY KEY,  
  company_id BIGINT NOT NULL REFERENCES companies(id),  
  
  order_date DATE NOT NULL,  
  
  -- Агрегация  
  total_items INT DEFAULT 0, -- Сумма порций  
  status VARCHAR(50) DEFAULT 'PLANNED', -- PLANNED → CONFIRMED →  
  HANDED_TO_LOGISTICS → DONE  
  
  -- Логистика  
  delivery_task_id BIGINT REFERENCES delivery_tasks(id), --  Связь с нов  
  ой логистикой  
  
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

## Модуль 5: Core Finance (Central Ledger)

Здесь мы считаем деньги Партнеров и Компаний. Личные кошельки физлиц (B2C) живут в **Client App** и сюда не попадают (только агрегированные суммы).

### core\_transactions (ex-Transactions)

Главная книга экосистемы.



```

CREATE TABLE core_transactions (
  id BIGSERIAL PRIMARY KEY,

  -- Субъект (Чей баланс меняем?)
  entity_type VARCHAR(50) NOT NULL, -- 'PARTNER', 'COMPANY'
  entity_id BIGINT NOT NULL,

  type VARCHAR(50) NOT NULL,
  -- 'B2B_INVOICE_PAYMENT' (+ Компания)
  -- 'B2B_LUNCH_CHARGE' (- Компания)
  -- 'PARTNER_PAYOUT' (- Партнер, вывод средств)
  -- 'PARTNER_REVENUE' (+ Партнер, выручка за заказы)

  amount DECIMAL(15, 2) NOT NULL,

  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

```

## Модуль 6: Access (System Users)

### **system\_users** (ex-Users)

Только внутренний персонал.

```

CREATE TABLE system_users (
  id BIGSERIAL PRIMARY KEY,
  name VARCHAR(150) NOT NULL,
  email VARCHAR(150) UNIQUE,
  password_hash VARCHAR(255),

  role VARCHAR(50) NOT NULL, -- 'SUPERADMIN', 'SUPPORT', 'LOGISTIC_M
  ANAGER', 'SALES'
  is_active BOOLEAN DEFAULT TRUE,

```

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

## 3. Интеграционная Карта (The Map)

Как Yalla CRM общается с внешним миром.

### 1. Yalla CRM <-> Business CRM

- **Направление:** Master → Slave.
- **Sync:**
  - Создание `companies` в Yalla CRM → Webhook → Создание `companies` в Business CRM.
  - Генерация `invoices` в Yalla CRM (Billing) → Webhook → Появление счета в Business CRM.
- **Legacy Flow:**
  - В 10:30 `Business CRM` отдает сводку заказов → `Yalla CRM` создает `daily_company_orders`.

### 2. Yalla CRM <-> Merchant CRM

- **Направление:** Master → Slave.
- **Sync:**
  - Создание `partners` в Yalla CRM → Webhook → Создание `stores` в Merchant CRM.
  - Обновление `global_categories` → Sync → Merchant CRM.
- **Logistics Flow:**
  - Merchant CRM кидает запрос "Нужен курьер" → Yalla CRM создает `delivery_tasks` (B2C) и назначает курьера. Статус курьера летит обратно по WebSocket.

### 3. Yalla CRM <-> Client App

- **Направление:** Support & Catalog.

- **Data:**

- Client App читает `global_products` и `partners` (через API Gateway) для отображения каталога.
- Client App шлет `incidents` (жалобы) → Yalla CRM Incident Queue.

## ⚠ 4. План Миграции (Migration Strategy)

Так как у нас уже есть данные, миграцию нужно проводить в 3 этапа.

### 1. Этап 1: Parallel Run (Двойная запись)

- Создать новые таблицы ( `partners` , `companies` , `couriers` ).
- Написать скрипт, который копирует данные из `ACTORS` в новые таблицы.
- Настроить триггеры: при вставке в `ACTORS` дублировать в новые таблицы.

### 2. Этап 2: Switch Reads (Переключение чтения)

- Переписать API эндпоинты на чтение из новых таблиц.
- `ACTORS` остается только для записи легаси-кода.

### 3. Этап 3: Switch Writes & Cleanup (Переключение записи)

- Переписать логику создания на новые таблицы.
- Удалить таблицу `ACTORS` и старый код.

## Резюме архитектора

Эта структура решает главную боль — **зависимость от монолитной таблицы ACTORS**. Мы выделили чистые домены (Логистика, Партнеры, B2B), сохранили работающий "обеденный" движок и подготовили почву для Ритейла через PIM-каталог.