

Merchant CRM

Техническая спецификация БД: Merchant CRM

Контекст: Рабочее место ресторана/магазина (Admin Panel + Tablet App).
"Пульс" всей системы доставки.

0. Словарь Сущностей (Entity Reference)

Список всех объектов базы данных с их ролью в экосистеме.

Сущность	Описание	Критичность
Store	Торговая точка (Ресторан, Даркстор, Магазин). Владелец меню и депозита.	 High
Merchant User	Сотрудник точки (Агент, Кассир, Владелец). "Руки" системы.	 Med
Shift	Рабочая смена. Определяет доступность магазина в Client App.	 High
Schedule	Плановое расписание. Используется роботом для авто-закрытия забытых смен.	 Low
Global Category	Справочник категорий Yalla. Общий для всех.	 Low
Menu Item	Товар/Блюдо. Локальная сущность точки. Содержит цену и тайминг.	 Med
Modifier	Опции товара (Размер, Топпинги). Локальная сущность.	 Low
Merchant Order	Заказ глазами производства (Кухни/Склада).	 High
Store Transaction	Запись в финансовой книге. Любое изменение депозита.	 High

1. ER-Диаграмма

```

erDiagram
    %% Core Store
    STORES ||--|{ MERCHANT_USERS : employs
    STORES ||--|{ SHIFTS : operates
    STORES ||--|{ STORE_SCHEDULES : has_plan
    STORES ||--|{ STORE_TRANSACTIONS : finances

    %% Menu / Catalog
    GLOBAL_CATEGORIES ||--|{ MENU_ITEMS : classifies
    STORES ||--|{ MENU_ITEMS : owns
    MENU_ITEMS ||--|{ ITEM_MODIFIERS : configurable_by
    MODIFIER_GROUPS ||--|{ ITEM_MODIFIERS : groups

    %% Orders (Production)
    STORES ||--|{ MERCHANT_ORDERS : fulfills
    MERCHANT_ORDERS ||--|{ ORDER_ITEMS : consists_of
    MERCHANT_USERS ||--o{ MERCHANT_ORDERS : processed_by

stores {
    bigint id PK
    string type "RESTAURANT/RETAIL"
    decimal deposit_balance
    string status "ACTIVE/BLOCKED"
}

merchant_orders {
    bigint id PK
    uuid client_order_uuid
    string status
    int cooking_time_actual
}

```

2. Детальная SQL Схема и Комментарии



Модуль 1: Store Core (Точка и Настройки)

stores

Единая сущность для Ресторанов и Ритейла. **Логика:** Это "Контейнер" для меню, денег и персонала.

```
CREATE TABLE stores (
    id BIGSERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    -- Тип бизнеса (Закладка под Ритейл)
    store_type VARCHAR(20) DEFAULT 'RESTAURANT', -- 'RESTAURANT', 'GROCERY', 'DARK_KITCHEN'
    -- Статусы
    status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE (Работает), BLOCKED_FINANCE (Долг), ARCHIVED (Закрылся)
    is_promoted BOOLEAN DEFAULT FALSE, -- Буст в выдаче Client App
    -- Финансы (Депозит)
    deposit_balance DECIMAL(15, 2) DEFAULT 0.00, -- Текущий баланс (может быть < 0)
    overdraft_limit DECIMAL(15, 2) DEFAULT 1000.00, -- Сколько разрешено уйти в минус
    currency_code CHAR(3) DEFAULT 'TJS',
    -- Коммерческие условия
    commission_type VARCHAR(20) DEFAULT 'COMMISSION', -- 'COMMISSION' (с заказа), 'SERVICE_FEE' (с клиента)
    commission_rate DECIMAL(5, 2) DEFAULT 0.00, -- % Yalla
    -- Логистика и Гео
    geo_point POINT NOT NULL, -- Координаты входа
    delivery_radius_meters INT DEFAULT 3000, -- Зона обслуживания
    address_text TEXT,
    timezone VARCHAR(50) DEFAULT 'Asia/Dushanbe', -- Важно для авто-закрытия смен
```

```
-- Метаданные
logo_url TEXT,
cover_url TEXT,

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

💡 Комментарии Архитектора:

- `store_type`: Фронтенд должен смотреть на это поле. Если `GROCERY` — показываем интерфейс сборщика (сканер ШК). Если `RESTAURANT` — интерфейс повара (таймеры).
- `deposit_balance`: Изменяется **только** через транзакции.
Прямой `UPDATE` запрещен конвенцией.

store_schedules

Плановый график работы. **Назначение:** Используется скриптом "System Auto-Close" для принудительного закрытия забытых смен.

```
CREATE TABLE store_schedules (
    id BIGSERIAL PRIMARY KEY,
    store_id BIGINT NOT NULL REFERENCES stores(id) ON DELETE CASCADE,
    day_of_week INT NOT NULL CHECK (day_of_week BETWEEN 1 AND 7), -- 1
    =Monday
    is_working_day BOOLEAN DEFAULT TRUE,
    open_time TIME,
    close_time TIME,
    -- Поддержкаочных смен (например, до 02:00 ночи следующего дня)
    closes_next_day BOOLEAN DEFAULT FALSE,
```

```
    UNIQUE (store_id, day_of_week)
);
```



Модуль 2: Staff & Shifts (Персонал и Смены)

merchant_users

Сотрудники точки. **Связь:** Вход по номеру телефона. Это локальные пользователи. Если нужно связать их с глобальным аккаунтом Yalla, это делается на уровне API Gateway.

```
CREATE TABLE merchant_users (
    id BIGSERIAL PRIMARY KEY,
    -- Auth
    phone VARCHAR(20) NOT NULL UNIQUE, -- Логин
    password_hash VARCHAR(255), -- Пароль (выдается админом)
    full_name VARCHAR(150),
    -- Роль
    role VARCHAR(50) NOT NULL, -- 'AGENT' (Линейный), 'OWNER' (Управляет деньгами), 'SUPER_AGENT' (Yalla Support)
    -- HR статусы
    status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE, BLOCKED (Уволен - мгновенный кик)
    is_yalla_staff BOOLEAN DEFAULT FALSE, -- Флаг для отчетности (чей сотрудник)
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);
```

merchant_user_bindings

Матрица доступа. Позволяет одному человеку работать в нескольких ресторанах (сетевой менеджер).

```
CREATE TABLE merchant_user_bindings (
    user_id BIGINT NOT NULL REFERENCES merchant_users(id) ON DELETE C
    ASCADE,
    store_id BIGINT NOT NULL REFERENCES stores(id) ON DELETE CASCADE,
    is_primary BOOLEAN DEFAULT FALSE, -- Ресторан по умолчанию при вхо
    де
    permissions JSONB DEFAULT '{}', -- Доп. права (напр. {"can_edit_menu": f
    alse})
    created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    PRIMARY KEY (user_id, store_id)
);
```

shifts (КРИТИЧНО)

Журнал операционной активности. **Бизнес-правило:** Ресторан виден в Client App ТОЛЬКО если есть активная смена.

```
CREATE TABLE shifts (
    id BIGSERIAL PRIMARY KEY,
    store_id BIGINT NOT NULL REFERENCES stores(id),
    user_id BIGINT NOT NULL REFERENCES merchant_users(id), -- Кто открыл
    status VARCHAR(50) DEFAULT 'ACTIVE', -- ACTIVE, PAUSED (Перерыв), C
    LOSED
    started_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    ended_at TIMESTAMP WITH TIME ZONE,
    -- Аудит закрытия
    closed_by_type VARCHAR(50), -- 'SELF', 'SYSTEM_AUTO_CLOSE', 'ADMIN_
    FORCE'
```

```
closed_by_user_id BIGINT, -- ID того, кто закрыл (если SELF или ADMIN)
```

```
-- Статистика (Snapshot при закрытии)
```

```
orders_processed INT DEFAULT 0,
```

```
total_revenue_session DECIMAL(15, 2) DEFAULT 0.00
```

```
);
```

```
-- Индекс Heartbeat: Главный запрос Client App ("Кто открыт?")
```

```
CREATE INDEX idx_shifts_active ON shifts(store_id) WHERE status = 'ACTIVE';
```



Модуль 3: Menu (Меню и Склад)

global_categories

Глобальный справочник Yalla. **Логика:** Рестораны не могут создавать свои категории, чтобы каталог в Client App был чистым.

```
CREATE TABLE global_categories (
    id SERIAL PRIMARY KEY,
    name_ru VARCHAR(100) NOT NULL,
    name_tj VARCHAR(100),
    name_en VARCHAR(100),
    icon_url TEXT,
    sort_order INT DEFAULT 0
);
```

menu_items

Товары / Блюда. **Владение:** Полностью управляется рестораном. **Ритейл:** Для магазина поле `cooking_time` означает "Время сборки заказа".

```
CREATE TABLE menu_items (
    id BIGSERIAL PRIMARY KEY,
    store_id BIGINT NOT NULL REFERENCES stores(id),
    category_id INT REFERENCES global_categories(id),
```

```

name VARCHAR(255) NOT NULL,
description TEXT,
image_url TEXT,

-- Экономика
price DECIMAL(10, 2) NOT NULL CHECK (price >= 0),
old_price DECIMAL(10, 2), -- Для отображения скидок

-- Логистика (Human API)
cooking_time_minutes INT NOT NULL DEFAULT 15, -- ! База для алгоритма вызова курьера

-- Наличие (Оперативное управление)
is_out_of_stock BOOLEAN DEFAULT FALSE, -- "Стоп-лист"
is_archived BOOLEAN DEFAULT FALSE, -- Удалено из меню

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

```

 Рекомендации:

- При изменении `price` или `name` не забывайте, что это не влияет на старые заказы (там хранятся снапшоты).

`modifier_groups` И `item_modifiers`

Локальные модификаторы. **Пример:** Группа "Размер" → Опции "S, M, L".

```

CREATE TABLE modifier_groups (
id BIGSERIAL PRIMARY KEY,
store_id BIGINT NOT NULL REFERENCES stores(id), -- Локальная группа ресторана
name VARCHAR(100) NOT NULL, -- "Соусы", "Степень прожарки"

min_selection INT DEFAULT 0, -- 0 = Опционально
max_selection INT DEFAULT 1, -- 1 = Radio button, >1 = Checkbox

```

```

    is_required BOOLEAN DEFAULT FALSE
);

CREATE TABLE item_modifiers (
    id BIGSERIAL PRIMARY KEY,
    group_id BIGINT NOT NULL REFERENCES modifier_groups(id) ON DELETE
CASCADE,
    name VARCHAR(100) NOT NULL, -- "Сырный", "Острый"
    price_delta DECIMAL(10, 2) DEFAULT 0.00, -- Наценка (+5.00 TJS)

    is_out_of_stock BOOLEAN DEFAULT FALSE -- Можно отключить конкретн
ый соус
);

-- Таблица связей Блюдо ↔ Группа
CREATE TABLE menu_item_modifier_groups (
    menu_item_id BIGINT REFERENCES menu_items(id) ON DELETE CASCADE,
    modifier_group_id BIGINT REFERENCES modifier_groups(id) ON DELETE CA
SCADE,
    sort_order INT DEFAULT 0,
    PRIMARY KEY (menu_item_id, modifier_group_id)
);

```

Модуль 4: Orders (Производство)

merchant_orders

Заказ в контексте производства. **Связь:** Привязан к глобальному заказу Client App через UUID.

```

CREATE TABLE merchant_orders (
    id BIGSERIAL PRIMARY KEY,
    store_id BIGINT NOT NULL REFERENCES stores(id),
    -- Внешняя связь (Client App)

```

```
client_app_order_uuid UUID NOT NULL, -- 🔗 Ссылка на глобальный заказ
client_phone VARCHAR(20), -- Для экстренной связи
client_name VARCHAR(150),
delivery_address_json JSONB, -- Снапшот адреса для курьера

-- Статусы (Production Pipeline)
status VARCHAR(50) DEFAULT 'NEW',
-- NEW (Звенит) → COOKING (Принят) → READY_FOR_PICKUP (Упакован)
→ HANDOVER (Передан курьеру) → CANCELED

-- Ответственность
accepted_by_user_id BIGINT REFERENCES merchant_users(id), -- Агент, на
жавший "Принять"

-- Тайминг (Auto-Logistics Data)
created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(), -- Время посту
пления
accepted_at TIMESTAMP WITH TIME ZONE, -- Старт готовки

estimated_cooking_time INT, -- План (мин)
target_ready_time TIMESTAMP WITH TIME ZONE, -- План готовности (для
вызыва курьера)
actual_ready_time TIMESTAMP WITH TIME ZONE, -- Факт готовности (наж
атие "Готово")

-- SOS / Проблемы
is_late BOOLEAN DEFAULT FALSE, -- Флаг нарушения SLA
problem_reason VARCHAR(100), -- "Сгорело", "Нет света"

-- Финансы (Снэпшот заказа)
total_amount DECIMAL(15, 2) NOT NULL, -- Сумма товаров + упаковки
commission_amount DECIMAL(15, 2) DEFAULT 0.00 -- Рассчитанная коми
сия Yalla
);

-- Индекс для ленты заказов (Active Orders)
```

```
CREATE INDEX idx_merchant_orders_active ON merchant_orders(store_id)
WHERE status IN ('NEW', 'COOKING', 'READY_FOR_PICKUP');
```

order_items (Snapshot)

Состав заказа. **Важно:** Полностью дублирует данные из меню на момент заказа. Защита от изменения цен.

```
CREATE TABLE order_items (
    id BIGSERIAL PRIMARY KEY,
    merchant_order_id BIGINT NOT NULL REFERENCES merchant_orders(id) O
N DELETE CASCADE,
    menu_item_id BIGINT, -- Ссылка может стать NULL, если блюдо удалят с
овсем
    -- ЧАПШОТЫ (Immutable Data)
    name_snapshot VARCHAR(255) NOT NULL,
    price_snapshot DECIMAL(10, 2) NOT NULL,
    quantity INT NOT NULL,
    -- Выбранные опции
    modifiers_json JSONB, -- [{"name": "Сыр", "price": 5.00}]
    total_price DECIMAL(10, 2) NOT NULL -- (price + mods) * qty
);
```

💰 Модуль 5: Finance (Леджер)

store_transactions

Финансовая история. Единственный способ изменить баланс магазина.

```
CREATE TABLE store_transactions (
    id BIGSERIAL PRIMARY KEY,
```

```

store_id BIGINT NOT NULL REFERENCES stores(id),

type VARCHAR(50) NOT NULL,
-- 'ORDER_COMMISSION' (Списание комиссии),
-- 'ORDER_REVENUE' (Зачисление выручки, если эквайринг наш),
-- 'QR_PAYMENT_IN' (Приход от гостя в зале),
-- 'DEPOSIT_TOPUP' (Пополнение депозита),
-- 'WITHDRAWAL' (Вывод средств)

amount DECIMAL(15, 2) NOT NULL, -- Отрицательное ( списание ) или Положительное (приход)
currency_code CHAR(3) DEFAULT 'TJS',

-- Traceability (Основание платежа)
merchant_order_id BIGINT REFERENCES merchant_orders(id),
external_ref_id VARCHAR(100), -- ID транзакции банка / QR

-- Integrity Check
balance_before DECIMAL(15, 2) NOT NULL,
balance_after DECIMAL(15, 2) NOT NULL, -- Должно сходиться: before + amount = after

created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

```

⚠ 3. Checklist для Разработчиков (Dev Rules)

🔴 Critical (Безопасность и Деньги)

- Ledger Integrity:** Запрещено делать `UPDATE stores SET deposit_balance = ...`. Баланс обновляется только через `INSERT` в `store_transactions` внутри транзакции.
- Shift Availability:** API `GET /restaurants` в Client App должно фильтровать рестораны запросом: `WHERE EXISTS (SELECT 1 FROM shifts WHERE store_id = stores.id AND status = 'ACTIVE')`. Нет смены = Нет ресторана.

3. **Optimistic Locking:** При приеме заказа (`status NEW → COOKING`) проверяйте, что статус всё еще `NEW`. Если два агента нажали одновременно, второй должен получить ошибку, а не перепринять заказ.

🟡 Warning (Логика)

1. **Auto-Logistics:** Бэкенд должен иметь Cron/Worker, который каждую минуту проверяет заказы в статусе `COOKING`. Если `NOW() >= target_ready_time - 15 min`, отправляется запрос в API Логистики на поиск курьера.
2. **QR Notifications:** QR оплата создает транзакцию и посыпает WebSocket event на планшет. Она **не** меняет статусы кухонных заказов (они независимы).

🔵 Retail Ready (Задел на будущее)

1. **Terminology:** В коде и интерфейсе используйте `store_type`.
 - Если `GROCERY : cooking_time` → "Время сборки". `COOKING status` → "ASSEMBLING".
 - Логика БД остается единой.