

Архитектор базы данных

1 Основные сущности

Таблица: users (Пользователи)

Назначение: хранение учетных данных, контактной информации и прав доступа.

- **Поля:**

- id (INTEGER) — уникальный идентификатор пользователя. PRIMARY KEY, генерируется последовательностью.
- name (VARCHAR) — отображаемое имя.
- email (VARCHAR(255)) — электронная почта (логин).
- phone (VARCHAR(32)) — телефон.
- password_hash (VARCHAR) — хэш пароля.
- is_admin (BOOLEAN) — флаг административных прав (доступ на добавление/удаление товаров).
- created_at, updated_at (TIMESTAMP TZ) — метки создания и обновления, created_at по умолчанию now().

- **Ограничения:**

- users_pk (PRIMARY KEY): id.
- NOT NULL: email, password_hash, is_admin, id, created_at.

- **Индексы:**

- users_email_key (UNIQUE INDEX, BTREE): уникальность email.

Таблица: product_categories (Категории товаров)

Назначение: классификация товарного каталога.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- name (VARCHAR) — название категории.
- description (TEXT) — описание.

- created_at, updated_at (TIMESTAMP TZ).

- **Ограничения:**

- product_categories_pk (PRIMARY KEY): id.
- NOT NULL: id, name, created_at.

- **Индексы:**

- product_categories_name_key (UNIQUE INDEX, BTREE): уникальность названий.

Таблица: products (Товары)

Назначение: карточки товаров охранных систем.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- sku (VARCHAR) — артикул.
- name (VARCHAR) — наименование.
- product_type (VARCHAR) — тип изделия (камера, датчик, комплект и т.п.).
- price (DECIMAL(12,2)) — цена.
- availability_status (VARCHAR) — статус наличия (in_stock, preorder и т.п.).
- category_id (INTEGER) — ссылка на product_categories.
- created_at, updated_at (TIMESTAMP TZ).

- **Ограничения:**

- products_pk (PRIMARY KEY): id.
- products_category_id_fkey (FOREIGN KEY): category_id → product_categories(id).
- NOT NULL: id, name, price, category_id, created_at.

- **Индексы:**

- products_sku_key (UNIQUE INDEX, BTREE): уникальность артикула.
- idx_products_category_id (INDEX, BTREE): ускорение фильтрации по категории.

Таблица: carts (Корзины)

Назначение: активная корзина пользователя.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- user_id (INTEGER) — владелец корзины (users).
- created_at (TIMESTAMP TZ).

- **Ограничения:**

- carts_pk (PRIMARY KEY): id.
- carts_user_id_fkey (FOREIGN KEY): user_id → users(id).
- NOT NULL: id, user_id, created_at.

- **Индексы:**

- carts_user_id_key (UNIQUE INDEX, BTREE): гарантирует одну корзину на пользователя.

Таблица: cart_items (Позиции корзины)

Назначение: товары, добавленные в корзину.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- cart_id (INTEGER) — ссылка на carts.
- product_id (INTEGER) — ссылка на products.
- quantity (INTEGER) — количество.
- price (DECIMAL(12,2)) — цена на момент добавления.
- created_at, updated_at (TIMESTAMP TZ).

- **Ограничения:**

- cart_items_pk (PRIMARY KEY): id.
- cart_items_cart_id_fkey (FOREIGN KEY): cart_id → carts(id).
- cart_items_product_id_fkey (FOREIGN KEY): product_id → products(id).
- NOT NULL: id, cart_id, product_id, quantity, created_at.

- **Индексы:**

- cart_items_pair_key (UNIQUE INDEX, BTREE): (cart_id, product_id) — запрещает дубли.
- idx_cart_items_cart_id (INDEX, BTREE).

Таблица: addresses (Адреса)

Назначение: адреса доставки и адреса объектов сервиса.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- city (VARCHAR), street (VARCHAR), house (VARCHAR), apartment (VARCHAR).
- comment (TEXT) — примечание.

- created_at, updated_at (TIMESTAMP TZ).

- **Ограничения:**

- addresses_pk (PRIMARY KEY): id.
- NOT NULL: id, city, street, house, created_at.

- **Индексы:**

- idx_addresses_city (INDEX, BTREE).

Таблица: service_objects (Объекты сервиса)

Назначение: места установки/обслуживания систем.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- name (VARCHAR) — название объекта.
- object_type (VARCHAR) — тип объекта (квартира, офис и т.п.).
- area (DECIMAL(12,2)) — площадь, м².
- address_id (INTEGER) — ссылка на addresses (1:1).
- created_at, updated_at (TIMESTAMP TZ).

- **Ограничения:**

- service_objects_pk (PRIMARY KEY): id.
- service_objects_address_id_fkey (FOREIGN KEY): address_id → addresses(id).
- NOT NULL: id, name, address_id, created_at.

- **Индексы:**

- service_objects_address_id_key (UNIQUE INDEX, BTREE): обеспечивает связь 1:1 с адресом.

Таблица: service_requests (Сервисные заявки)

Назначение: учет заявок на установку и ремонт.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- number (VARCHAR) — номер заявки.
- type (VARCHAR) — тип: установка/ремонт.
- status (VARCHAR) — статус исполнения.
- description (TEXT) — описание проблемы/работ.
- service_object_id (INTEGER) — ссылка на service_objects.

- created_at, updated_at (TIMESTAMPTZ).

- **Ограничения:**

- service_requests_pk (PRIMARY KEY): id.
- service_requests_service_object_id_fkey (FOREIGN KEY): service_object_id → service_object_id.
- NOT NULL: id, number, type, status, service_object_id, created_at.

- **Индексы:**

- service_requests_number_key (UNIQUE INDEX, BTREE).
- idx_service_requests_service_object_id (INDEX, BTREE).
- idx_service_requests_status (INDEX, BTREE).

Таблица: install_requests (Заявки на установку)

Назначение: специализация service_requests для установок.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- service_request_id (INTEGER) — заявка-родитель.
- date (DATE) — дата установки.
- created_at, updated_at (TIMESTAMPTZ).

- **Ограничения:**

- install_requests_pk (PRIMARY KEY): id.
- install_requests_service_request_id_fkey (FOREIGN KEY): service_request_id → service_requests(id).
- NOT NULL: id, service_request_id, date, created_at.

- **Индексы:**

- install_requests_service_request_id_key (UNIQUE INDEX, BTREE): связь 1:1 с заявкой.

Таблица: repair_requests (Заявки на ремонт)

Назначение: специализация service_requests для ремонтов. Структура аналогична install_requests.

- **Ключевые ограничения и индексы:** PRIMARY KEY id; FOREIGN KEY service_request_id → service_requests(id); UNIQUE INDEX по service_request_id; NOT NULL для id, service_request_id, date, created_at.

Таблица: estimates (Сметы)

Назначение: согласование стоимости работ и материалов по заявке.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- service_request_id (INTEGER) — ссылка на заявку.
- labor_cost (DECIMAL(12,2)) — стоимость работ.
- material_cost (DECIMAL(12,2)) — стоимость материалов.
- status (VARCHAR) — статус сметы.
- created_at, updated_at (TIMESTAMP TZ).

- **Ограничения:**

- estimates_pk (PRIMARY KEY): id.
- estimates_service_request_id_fkey (FOREIGN KEY): service_request_id → service_requests(id)
- NOT NULL: id, service_request_id, labor_cost, material_cost, created_at.

- **Индексы:**

- estimates_service_request_id_key (UNIQUE INDEX, BTREE): 1:1 со служебной заявкой.

Таблица: notifications (Уведомления)

Назначение: сообщения по ходу обработки заявок.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- service_request_id (INTEGER) — ссылка на заявку.
- text (TEXT) — содержимое уведомления.
- sent_at (TIMESTAMP TZ) — время отправки.

- **Ограничения:**

- notifications_pk (PRIMARY KEY): id.
- notifications_service_request_id_fkey (FOREIGN KEY): service_request_id → service_requests(id)
- NOT NULL: id, service_request_id, text, sent_at.

- **Индексы:**

- idx_notifications_request_sent (INDEX, BTREE): (service_request_id, sent_at DESC).

Таблица: orders (Заказы)

Назначение: коммерческие заказы, сформированные по результатам заявок.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- number (VARCHAR) — номер заказа.
- created_at (TIMESTAMPTZ) — дата оформления.
- status (VARCHAR) — статус заказа.
- total_amount (DECIMAL(12,2)) — сумма заказа.
- service_request_id (INTEGER) — связь с service_requests (обычно 1:1).
- address_id (INTEGER) — адрес выполнения/доставки.
- updated_at (TIMESTAMPTZ).

- **Ограничения:**

- orders_pk (PRIMARY KEY): id.
- orders_service_request_id_fkey (FOREIGN KEY): service_request_id → service_requests(id).
- orders_address_id_fkey (FOREIGN KEY): address_id → addresses(id).
- NOT NULL: id, number, created_at, total_amount.

- **Индексы:**

- orders_number_key (UNIQUE INDEX, BTREE).
- idx_orders_created_at (INDEX, BTREE).
- orders_service_request_id_key (UNIQUE INDEX, BTREE): обеспечивает связь 1:1 с заявкой.

Таблица: deliveries (Доставки)

Назначение: параметры доставки по заказу.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- order_id (INTEGER) — ссылка на orders.
- address_id (INTEGER) — адрес доставки.
- type (VARCHAR) — тип доставки (курьер, самовывоз).
- cost (DECIMAL(12,2)) — стоимость.
- eta (TIMESTAMPTZ) — планируемая дата/время.

- **Ограничения:**

- deliveries_pk (PRIMARY KEY): id.

- deliveries_order_id_fkey (FOREIGN KEY): order_id → orders(id).
- deliveries_address_id_fkey (FOREIGN KEY): address_id → addresses(id).
- NOT NULL: id, order_id, type, cost.

- **Индексы:**

- deliveries_order_id_key (UNIQUE INDEX, BTREE): связь 1:1 с заказом.

Таблица: payments (Платежи)

Назначение: движения оплат по заказам.

- **Поля:**

- id (INTEGER) — PRIMARY KEY.
- order_id (INTEGER) — ссылка на orders.
- method (VARCHAR) — метод оплаты.
- amount (DECIMAL(12,2)) — сумма.
- status (VARCHAR) — состояние («Зачислено», «Ожидает» и т.д.).
- paid_at (TIMESTAMPTZ) — время проведения, может быть NULL.
- created_at, updated_at (TIMESTAMPTZ).

- **Ограничения:**

- payments_pk (PRIMARY KEY): id.
- payments_order_id_fkey (FOREIGN KEY): order_id → orders(id).
- NOT NULL: id, order_id, method, amount, status, created_at.

- **Индексы:**

- idx_payments_order_id (INDEX, BTREE).
- idx_payments_status (INDEX, BTREE).

2 Связующие таблицы и специализации

Связующая: cart_items

Реализует связь «многие-ко-многим» между **cards** и **products**. Составной уникальный индекс (cart_id, product_id) предотвращает дубли одной и той же позиции.

Специализации service_requests

Таблицы **install_requests** и **repair_requests** находятся в отношении 1:1 с **service_requests** и расширяют заявку типоспецифичными полями (дата работ).

1:1 СВЯЗИ

- `users` — `carts`: уникальный `user_id` в корзине.
- `service_objects` — `addresses`: уникальный `address_id` в `service_objects`.
- `service_requests` — `estimates`, `orders`, `install_requests`, `repair_requests`.
- `orders` — `deliveries`.

3 Порядок заполнения и бизнес-правила

Порядок загрузки справочников и данных

1. `product_categories`
2. `products`
3. `users` (включая признак `is_admin`)
4. `addresses`, затем `service_objects`
5. `service_requests` и специализации (`install_requests/repair_requests`), далее `estimates`, `notifications`
6. `carts` и `cart_items`
7. `orders` → `deliveries`, `payments`

Ключевые бизнес-ограничения

- Один пользователь — одна активная корзина (`carts.user_id` уникален).
- В корзине не может быть двух строк с одним и тем же товаром (`cart_items(cart_id, product_id)` уникальны).
- Артикул товара (`products.sku`) и название категории (`product_categories.name`) уникальны.
- Администратор (`users.is_admin = TRUE`) имеет право создавать, изменять и удалять записи в `products`; для прочих пользователей операции ограничены политиками доступа.

Рекомендованные индексы производительности

- `orders(created_at)`, `payments(order_id, status)`, `notifications(service_request_id, sent_at)`.
- Индексы по всем внешним ключам для ускорения JOIN.