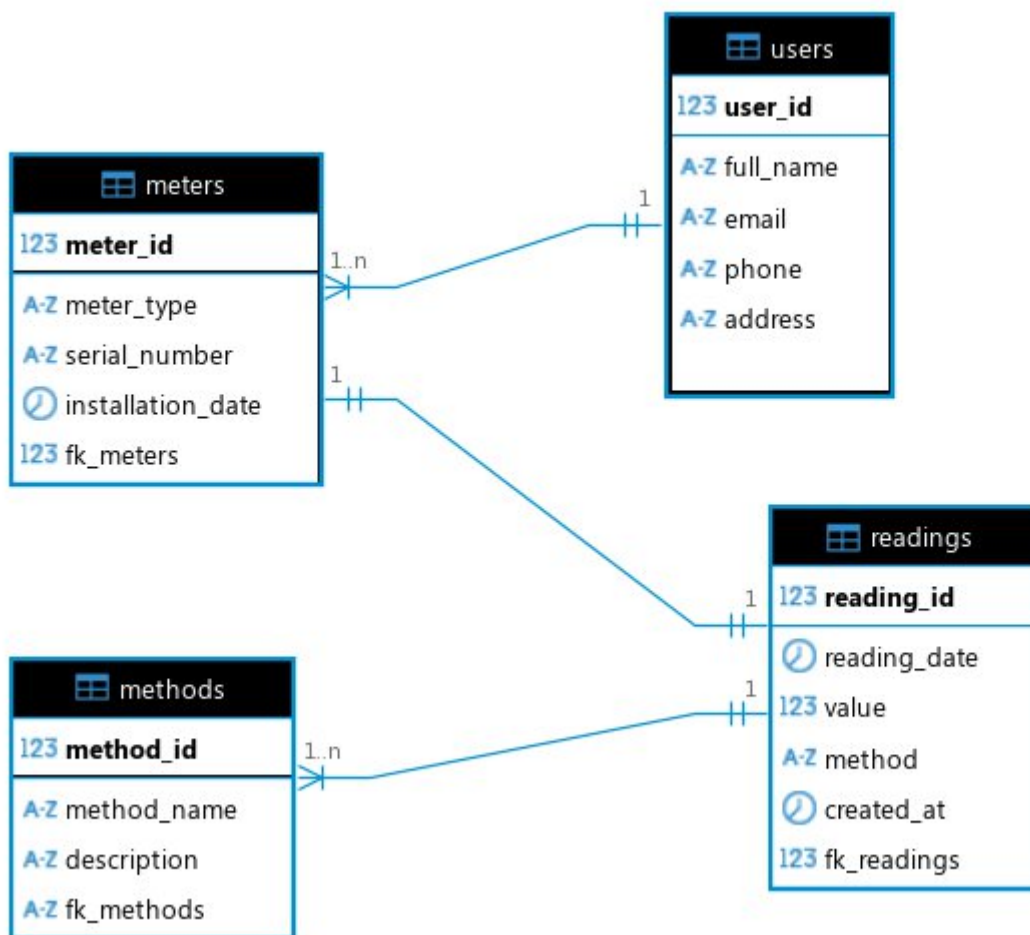


Бытовой кейс «Передача показаний счетчиков разными способами»

Для реализации кейса "Передача показаний счетчиков разными способами" в базе данных можно спроектировать следующую структуру сущностей и атрибутов. Основные сущности включают пользователей, счетчики, показания и способы передачи данных.



| Сущности и их атрибуты

| 1. Пользователи (users)

Сущность для хранения информации о пользователях, которые передают показания счетчиков.

```
CREATE TABLE users (
```

```
    user_id SERIAL PRIMARY KEY,    -- Уникальный идентификатор пользователя
```

```
    full_name VARCHAR(255) NOT NULL, -- Полное имя пользователя
```

```
email VARCHAR(255),          -- Электронная почта
phone VARCHAR(20),           -- Номер телефона
address TEXT                  -- Адрес пользователя
);
```

2. Счетчики (meters)

Сущность для хранения информации о счетчиках, которые принадлежат пользователям.

```
CREATE TABLE meters (
    meter_id SERIAL PRIMARY KEY, -- Уникальный идентификатор счетчика
    user_id INT NOT NULL REFERENCES users(user_id), -- Владелец счетчика (связь с
таблицей users)
    meter_type VARCHAR(50) NOT NULL, -- Тип счетчика (например, "электричество",
"вода", "газ")
    serial_number VARCHAR(100) NOT NULL UNIQUE, -- Серийный номер счетчика
    installation_date DATE        -- Дата установки счетчика
);
```

3. Показания (readings)

Сущность для хранения переданных показаний счетчиков.

```
CREATE TABLE readings (
    reading_id SERIAL PRIMARY KEY, -- Уникальный идентификатор показания
    meter_id INT NOT NULL REFERENCES meters(meter_id), -- Ссылка на счетчик (связь с
таблицей meters)
    reading_date TIMESTAMP NOT NULL, -- Дата и время передачи показаний
    value NUMERIC(10, 2) NOT NULL, -- Значение показания
    transmission_method VARCHAR(50) NOT NULL, -- Способ передачи данных (например,
"личный кабинет", "по телефону", "смс")
    created_at TIMESTAMP DEFAULT NOW() -- Дата и время записи в БД
);
```

4. Способы передачи данных (transmission_methods)

Сущность для хранения списка доступных способов передачи данных.

```
CREATE TABLE transmission_methods (
    method_id SERIAL PRIMARY KEY, -- Уникальный идентификатор способа передачи
```

```
method_name VARCHAR(50) NOT NULL UNIQUE, -- Название способа передачи
(например, "личный кабинет", "по телефону", "смс")
```

```
description TEXT -- Описание способа передачи
```

```
);
```

| Пример связей между таблицами

- Пользователь может иметь несколько счетчиков (связь users → meters).
- Один счетчик может иметь множество показаний (связь meters → readings).
- Каждое показание связано с конкретным способом передачи данных (связь readings → transmission_methods).

| Пример заполнения данных

| Добавление пользователей:

```
INSERT INTO users (full_name, email, phone, address)
```

```
VALUES
```

```
('Иван Иванов', 'ivanov@example.com', '+79991234567', 'Москва, ул. Ленина, д. 1'),
```

```
('Анна Смирнова', 'smirnova@example.com', '+79997654321', 'Санкт-Петербург, ул. Пушкина, д. 10');
```

| Добавление счетчиков:

```
INSERT INTO meters (user_id, meter_type, serial_number, installation_date)
```

```
VALUES
```

```
(1, 'электричество', '1234567890', '2020-01-01'),
```

```
(1, 'вода', '0987654321', '2021-06-15'),
```

```
(2, 'газ', '5678901234', '2019-09-10');
```

| Добавление способов передачи данных:

```
INSERT INTO transmission_methods (method_name, description)
```

VALUES

('личный кабинет', 'Передача через веб-интерфейс'),

('по телефону', 'Передача через звонок оператору'),

('смс', 'Передача через текстовое сообщение');

┃ Добавление показаний:

INSERT INTO readings (meter_id, reading_date, value, transmission_method)

VALUES

(1, '2023-10-01 12:00:00', 150.75, 'личный кабинет'),

(2, '2023-10-01 13:00:00', 300.50, 'смс'),

(3, '2023-10-01 14:00:00', 120.00, 'по телефону');

┃ Расширение модели

В зависимости от требований кейса можно добавить дополнительные таблицы или атрибуты:

- История изменения данных.
- Логи ошибок при передаче показаний.
- Статусы обработки показаний (например, "принято", "ошибка").
- Связь с тарифами для расчета стоимости потребления.

Бизнес-кейс "Управление складом и поставками"

Основные сущности и их описание:

1. **Товары (Products)**

Хранит информацию о товарах, которые находятся на складе или поставляются.

- Поля:

- `product_id` (уникальный идентификатор товара)
- `name` (название товара)
- `description` (описание товара)
- `category` (категория товара)
- `unit_price` (цена за единицу)
- `unit_of_measurement` (единица измерения, например, "шт", "кг", "л")
- `created_at` (дата добавления товара)

2. **Складские запасы (Inventory)**

Хранит текущее количество товаров на складе.

- Поля:

- `inventory_id` (уникальный идентификатор записи)
- `product_id` (ссылка на товар)
- `quantity_in_stock` (количество на складе)
- `warehouse_id` (ссылка на склад, если складов несколько)
- `last_updated_at` (дата последнего обновления)

3. **Склады (Warehouses)**

Хранит информацию о складах.

- Поля:

- `warehouse_id` (уникальный идентификатор склада)
- `name` (название склада)
- `location` (местоположение склада)

- `manager_name` (имя менеджера склада)

4. ****Поставщики (Suppliers)****

Хранит информацию о поставщиках, которые доставляют товары на склад.

- Поля:

- `supplier_id` (уникальный идентификатор поставщика)
- `name` (название компании-поставщика)
- `contact_person` (контактное лицо)
- `phone` (телефон поставщика)
- `email` (электронная почта поставщика)
- `address` (адрес поставщика)

5. ****Поставки (Deliveries)****

Хранит данные о поставках товаров от поставщиков на склад.

- Поля:

- `delivery_id` (уникальный идентификатор поставки)
- `supplier_id` (ссылка на поставщика)
- `warehouse_id` (ссылка на склад, куда была доставлена поставка)
- `delivery_date` (дата поставки)
- `total_cost` (общая стоимость поставки)

6. ****Позиции в поставке (Delivery_Items)****

Хранит данные о конкретных товарах в рамках одной поставки.

- Поля:

- `delivery_item_id` (уникальный идентификатор позиции в поставке)
- `delivery_id` (ссылка на поставку)
- `product_id` (ссылка на товар)
- `quantity_delivered` (количество доставленного товара)
- `cost_per_unit` (цена за единицу товара)

7. ****Отгрузки со склада (Shipments)****

Хранит данные об отправке товаров со склада клиентам или другим складам.

- Поля:

- `shipment_id` (уникальный идентификатор отгрузки)
- `warehouse_id` (ссылка на склад, с которого была сделана отгрузка)
- `shipment_date` (дата отгрузки)
- `destination` (место назначения отгрузки: адрес клиента или другого склада)

8. ****Позиции в отгрузке (Shipment_Items)****

Хранит данные о конкретных товарах в рамках одной отгрузки.

- Поля:

- `shipment_item_id` (уникальный идентификатор позиции в отгрузке)
- `shipment_id` (ссылка на отгрузку)
- `product_id` (ссылка на товар)
- `quantity_shipped` (количество отправленного товара)

Связи между таблицами

1. ****Products ↔ Inventory****: Один товар может быть на нескольких складах, поэтому связь один-ко-многим (`product_id` → `inventory.product_id`).
2. ****Warehouses ↔ Inventory****: Один склад может содержать множество товаров, связь один-ко-многим (`warehouse_id` → `inventory.warehouse_id`).
3. ****Suppliers ↔ Deliveries****: Один поставщик может сделать несколько поставок, связь один-ко-многим (`supplier_id` → `deliveries.supplier_id`).
4. ****Deliveries ↔ Delivery_Items****: Одна поставка может включать множество товаров, связь один-ко-многим (`delivery_id` → `delivery_items.delivery_id`).
5. ****Shipments ↔ Shipment_Items****: Одна отгрузка может включать множество товаров, связь один-ко-многим (`shipment_id` → `shipment_items.shipment_id`).
6. ****Products ↔ Delivery_Items / Shipment_Items****: Один товар может быть частью множества поставок и отгрузок, связь один-ко-многим (`product_id` → `delivery_items.product_id`, аналогично для `shipment_items`).