

```

- CREATE TABLE Users (
    userID SERIAL,
    emailAddress VARCHAR(40) NOT NULL UNIQUE,
    passwordHashSalt VARCHAR(40) NOT NULL,
    passwordHash VARCHAR(128) NOT NULL,
    name VARCHAR(40),
    surname VARCHAR(40),
    postalCode VARCHAR(40),
    city VARCHAR(40),
    street VARCHAR(40),
    houseNumber VARCHAR(40),
    token VARCHAR(40) UNIQUE,
    tokenCreation TIMESTAMP,
    userLendPeriod INTERVAL,
    lendStatus USERLENDSTATUS NOT NULL DEFAULT 'disabled',
    verificationStatus USERVERIFICATIONSTATUS NOT NULL DEFAULT 'unverified',
    userRole USERROLE NOT NULL DEFAULT 'registered',

    PRIMARY KEY(userID),
    CONSTRAINT positive_timestamp CHECK (tokenCreation >= NOW())

);

- CREATE OR REPLACE FUNCTION token_timestamp() RETURNS TRIGGER AS
$$
    BEGIN
        IF NEW.token != OLD.token THEN
            NEW.tokenCreation = NOW();
        END IF;
        RETURN NEW;
    END;
$$
LANGUAGE plpgsql;

- CREATE TRIGGER set_token_timestamp
    BEFORE UPDATE OF token ON Users
    FOR EACH ROW
    EXECUTE PROCEDURE token_timestamp();

```