Tabelas do modelo relacional

Por padrão, o Django cria automaticamente as tabelas do banco de dados durante a execução das migrações, utilizando a biblioteca postgresql_psycopg2 do Python. No entanto, podemos representá-las no PostgreSQL da seguinte forma:

Tabela de Fotografo -

```
Unset
CREATE TABLE Photographer (
    id_user SERIAL PRIMARY KEY,
    email VARCHAR(80) UNIQUE NOT NULL,
    senha VARCHAR(256) NOT NULL,
    name_photographer VARCHAR(60) NOT NULL,
    name_company VARCHAR(50),
    phone VARCHAR(20),
    s3_folder VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

Tabela de Album -

```
Unset
CREATE TABLE Album (
    id_album SERIAL PRIMARY KEY,
    id_fotografo INTEGER NOT NULL,
    name_album VARCHAR(50) NOT NULL,
    s3_path_album VARCHAR(256),
    created_at_album TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at_album TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (id_fotografo) REFERENCES Fotografo(id_usuario) ON DELETE
CASCADE
);
```

Tabela de Foto -

```
Unset
CREATE TABLE Photo (
    id_photo SERIAL PRIMARY KEY,
    id_album INTEGER NOT NULL,
    name_photo VARCHAR(60) NOT NULL,
    s3_path_photo VARCHAR(500) NOT NULL,
    format_photo VARCHAR(10),
    size_photo INTEGER,
    FOREIGN KEY (id_album) REFERENCES Album(id_album) ON DELETE CASCADE
);
```

Tabela de Tag -

```
Unset
CREATE TABLE Tag (
    id_tag SERIAL PRIMARY KEY,
    id_fotografo INTEGER NOT NULL,
    name_tag VARCHAR(20) NOT NULL,
    FOREIGN KEY (id_fotografo) REFERENCES Fotografo(id_usuario) ON DELETE
CASCADE
);
```

Tabela fruto da relação m:m de Album e Tag-

```
Unset

CREATE TABLE AlbumTag (
    id_tag INTEGER NOT NULL,
    id_album INTEGER NOT NULL,
    PRIMARY KEY (id_tag, id_album),
    FOREIGN KEY (id_tag) REFERENCES Tag(id_tag) ON DELETE CASCADE,
    FOREIGN KEY (id_album) REFERENCES Album(id_album) ON DELETE CASCADE
);
```