

```
-- =====
-- DDL matching current SQLAlchemy models (PostgreSQL)
-- =====

-- ----- ENUM TYPES -----
DO $$ BEGIN
    CREATE TYPE user_role AS ENUM ('consultant', 'company', 'admin');
EXCEPTION
    WHEN duplicate_object THEN null;
END $$;

DO $$ BEGIN
    CREATE TYPE unlock_target AS ENUM ('consultant', 'job');
EXCEPTION
    WHEN duplicate_object THEN null;
END $$;

DO $$ BEGIN
    CREATE TYPE collaboration_status AS ENUM ('active', 'ended');
EXCEPTION
    WHEN duplicate_object THEN null;
END $$;

-- ----- USERS -----
CREATE TABLE IF NOT EXISTS users (
    id      INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    username  VARCHAR(80) NOT NULL UNIQUE,
    role     user_role NOT NULL,
    created_at  TIMESTAMPTZ NOT NULL DEFAULT now()
);
```

```
-- Optional index (unique already creates an index on username)
-- CREATE INDEX IF NOT EXISTS idx_users_username ON users(username);
```

```
-- ----- COMPANIES -----
```

```
CREATE TABLE IF NOT EXISTS companies (
    id          INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    user_id     INT NOT NULL,
    company_name_masked VARCHAR(160) NOT NULL,
    industries   VARCHAR(255),
    location_city VARCHAR(120),
    country      VARCHAR(120),
    created_at    TIMESTAMPTZ NOT NULL DEFAULT now(),
    contact_email  VARCHAR(255),
    phone_number   VARCHAR(50),

    CONSTRAINT fk_companies_user
        FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

```
CREATE INDEX IF NOT EXISTS idx_companies_user_id ON companies(user_id);
```

```
-- ----- CONSULTANT PROFILES -----
```

```
CREATE TABLE IF NOT EXISTS consultant_profiles (
    id          INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    user_id     INT NOT NULL,
    display_name_masked VARCHAR(120) NOT NULL,
    headline     VARCHAR(160),
```

```

location_city      VARCHAR(120),
country           VARCHAR(120),
availability      BOOLEAN NOT NULL DEFAULT TRUE,
created_at        TIMESTAMPTZ NOT NULL DEFAULT now(),
years_experience  INT,

profile_image     VARCHAR(300),
cv_document       VARCHAR(300),
contact_email    VARCHAR(255),
phone_number     VARCHAR(50),

current_company_id INT,
latitude          DOUBLE PRECISION,
longitude         DOUBLE PRECISION,

CONSTRAINT fk_consultant_profiles_user
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,

CONSTRAINT fk_consultant_profiles_current_company
FOREIGN KEY (current_company_id) REFERENCES companies(id) ON DELETE SET NULL
);

CREATE INDEX IF NOT EXISTS idx_consultant_profiles_user_id ON consultant_profiles(user_id);
CREATE INDEX IF NOT EXISTS idx_consultant_profiles_current_company_id ON
consultant_profiles(current_company_id);

```

#### ----- SKILLS -----

```

CREATE TABLE IF NOT EXISTS skills (
id      INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
name    VARCHAR(120) NOT NULL UNIQUE,

```

```

created_at TIMESTAMPTZ NOT NULL DEFAULT now()
);

----- PROFILE_SKILLS (M2M) -----
CREATE TABLE IF NOT EXISTS profile_skills (
    profile_id INT NOT NULL,
    skill_id INT NOT NULL,
    created_at TIMESTAMPTZ NOT NULL DEFAULT now(),
    PRIMARY KEY (profile_id, skill_id),
    CONSTRAINT fk_profile_skills_profile
        FOREIGN KEY (profile_id) REFERENCES consultant_profiles(id) ON DELETE CASCADE,
    -- matches your SQLAlchemy: ondelete="RESTRICT"
    CONSTRAINT fk_profile_skills_skill
        FOREIGN KEY (skill_id) REFERENCES skills(id) ON DELETE RESTRICT
);

CREATE INDEX IF NOT EXISTS idx_profile_skills_profile_id ON profile_skills(profile_id);
CREATE INDEX IF NOT EXISTS idx_profile_skills_skill_id ON profile_skills(skill_id);

----- JOB POSTS -----
CREATE TABLE IF NOT EXISTS job_posts (
    id          INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    company_id   INT NOT NULL,
    title        VARCHAR(200) NOT NULL,
    description   TEXT,
    location_city VARCHAR(120),

```

```
country      VARCHAR(120),
contract_type  VARCHAR(80),
created_at    TIMESTAMPTZ NOT NULL DEFAULT now(),

is_active     BOOLEAN NOT NULL DEFAULT TRUE,
hired_consultant_id INT,

latitude      DOUBLE PRECISION,
longitude     DOUBLE PRECISION,

CONSTRAINT fk_job_posts_company
FOREIGN KEY (company_id) REFERENCES companies(id) ON DELETE CASCADE,

CONSTRAINT fk_job_posts_hired_consultant
FOREIGN KEY (hired_consultant_id) REFERENCES consultant_profiles(id) ON DELETE SET NULL
);

CREATE INDEX IF NOT EXISTS idx_job_posts_company_id ON job_posts(company_id);
CREATE INDEX IF NOT EXISTS idx_job_posts_hired_consultant_id ON job_posts(hired_consultant_id);
CREATE INDEX IF NOT EXISTS idx_job_posts_is_active ON job_posts(is_active);
```

-- ----- JOB\_SKILLS (M2M) -----

```
CREATE TABLE IF NOT EXISTS job_skills (
job_id    INT NOT NULL,
skill_id   INT NOT NULL,
created_at TIMESTAMPTZ NOT NULL DEFAULT now(),

PRIMARY KEY (job_id, skill_id),

CONSTRAINT fk_job_skills_job
```

```
    FOREIGN KEY (job_id) REFERENCES job_posts(id) ON DELETE CASCADE,  
  
    -- matches your SQLAlchemy: ondelete="RESTRICT"  
    CONSTRAINT fk_job_skills_skill  
        FOREIGN KEY (skill_id) REFERENCES skills(id) ON DELETE RESTRICT  
);
```

```
CREATE INDEX IF NOT EXISTS idx_job_skills_job_id ON job_skills(job_id);  
CREATE INDEX IF NOT EXISTS idx_job_skills_skill_id ON job_skills(skill_id);
```

```
-- ----- UNLOCKS -----
```

```
CREATE TABLE IF NOT EXISTS unlocks (  
    id      INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    user_id  INT NOT NULL,  
    target_type unlock_target NOT NULL,  
    target_id  INT NOT NULL,  
    created_at TIMESTAMP NOT NULL DEFAULT now(),
```

```
    CONSTRAINT fk_unlocks_user  
        FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE  
);
```

```
CREATE INDEX IF NOT EXISTS idx_unlocks_user_id ON unlocks(user_id);  
CREATE INDEX IF NOT EXISTS idx_unlocks_target ON unlocks(target_type, target_id);
```

```
-- Optional (recommended): prevent duplicate unlocks for same user/target  
-- CREATE UNIQUE INDEX IF NOT EXISTS uq_unlocks_user_target  
-- ON unlocks(user_id, target_type, target_id);
```

```
-- ----- COLLABORATIONS -----
```

```
CREATE TABLE IF NOT EXISTS collaborations (
    id          INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    company_id   INT NOT NULL,
    consultant_id INT NOT NULL,
    job_post_id   INT,
    status       collaboration_status NOT NULL DEFAULT 'active',
    started_at    TIMESTAMPTZ NOT NULL DEFAULT now(),
    ended_at     TIMESTAMPTZ,
    CONSTRAINT fk_collaborations_company
        FOREIGN KEY (company_id) REFERENCES companies(id) ON DELETE CASCADE,
```

```
CONSTRAINT fk_collaborations_consultant
```

```
    FOREIGN KEY (consultant_id) REFERENCES consultant_profiles(id) ON DELETE CASCADE,
```

```
CONSTRAINT fk_collaborations_job_post
```

```
    FOREIGN KEY (job_post_id) REFERENCES job_posts(id) ON DELETE SET NULL
```

```
);
```

```
CREATE INDEX IF NOT EXISTS idx_collaborations_company_id ON collaborations(company_id);
```

```
CREATE INDEX IF NOT EXISTS idx_collaborations_consultant_id ON collaborations(consultant_id);
```

```
CREATE INDEX IF NOT EXISTS idx_collaborations_job_post_id ON collaborations(job_post_id);
```

```
CREATE INDEX IF NOT EXISTS idx_collaborations_status ON collaborations(status);
```