# ERD — Entity Relationship Diagram

**Entities**

**Tenant**

Represents an organization using your API.

**UserAccount**

Represents individual users within a tenant.

**ApiKey**

Represents API keys used by external clients or services.
Required for *per API key* limits.

**ModelTier**

Represents model performance/price class (e.g., GPT-4, GPT-4-mini).
Required for *per model tier* limits.

**Model**

Represents actual base model used in inference (e.g., gpt-4, gpt-3.5).

**RateLimitPolicy**

Flexible policy table supporting:

- GLOBAL rules
- TENANT-level rules
- API_KEY-level rules
- MODEL-level rules
- MODEL_TIER-level rules
- USER_MODEL rules

Using the policy_scope enum.

**ERD Diagram**

Tenant (1) ——— (N) UserAccount

Tenant (1) ——— (N) ApiKey

ModelTier (1) ——— (N) Model

RateLimitPolicy:

scope: GLOBAL / TENANT / API_KEY / MODEL / MODEL_TIER / USER_MODEL

tenant_id FK

user_id FK

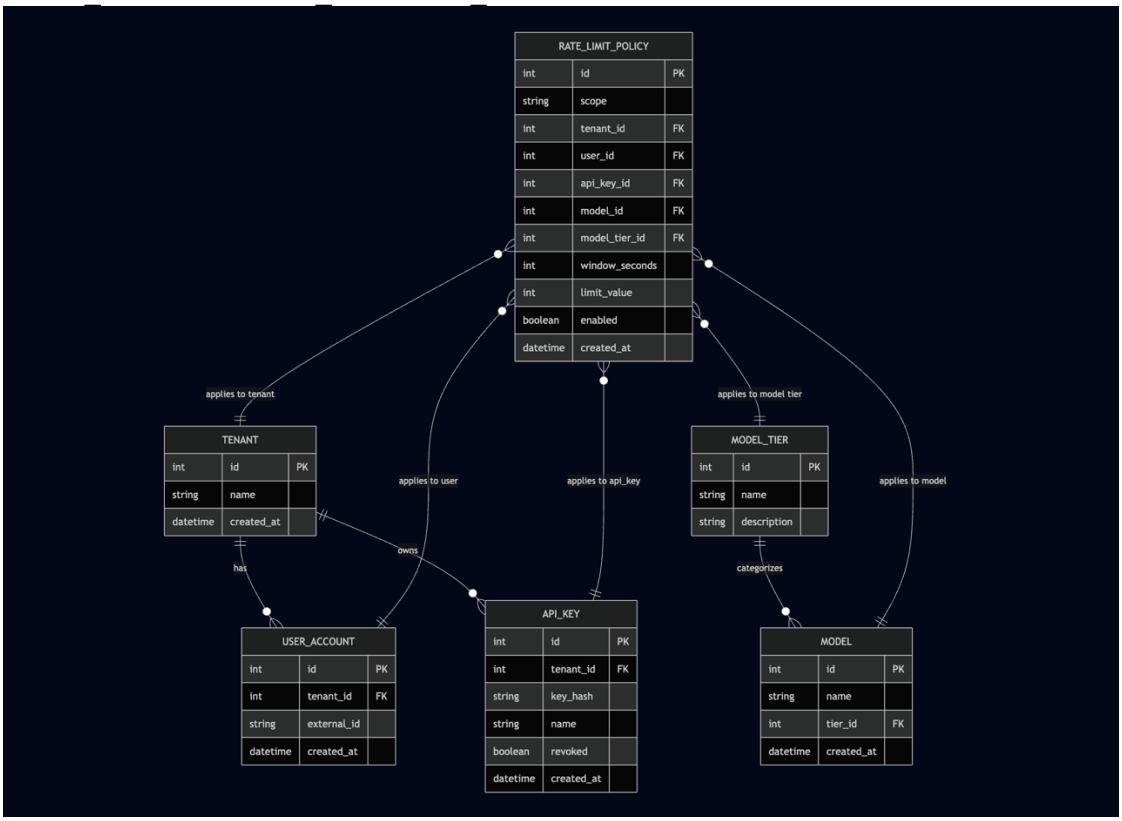
api_key_id FK

model_id FK

model_tier_id FK

Rate policies apply depending on scope:

- GLOBAL → no FK used
- TENANT → tenant_id
- API_KEY → api_key_id
- MODEL → model_id
- MODEL_TIER → model_tier_id
- USER_MODEL → user_id + model_id



### 4.1 Tables

**1. tenant**

```
CREATE TABLE tenant (
  id SERIAL PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  created_at TIMESTAMP NOT NULL DEFAULT NOW()
);
```

**2. user_account**

```
CREATE TABLE user_account (
  id SERIAL PRIMARY KEY,
  tenant_id INTEGER NOT NULL REFERENCES tenant(id),
  external_id VARCHAR(255),
  created_at TIMESTAMP NOT NULL DEFAULT NOW()
);
```

**3. api_key**

```
CREATE TABLE api_key (
  id SERIAL PRIMARY KEY,
  tenant_id INTEGER NOT NULL REFERENCES tenant(id),
  key_hash VARCHAR(255) NOT NULL UNIQUE,
  name VARCHAR(100),
  created_at TIMESTAMP NOT NULL DEFAULT NOW(),
  revoked BOOLEAN NOT NULL DEFAULT FALSE
);
```

This allows:

- Per API key rate limits
- API key grouping under a tenant

**4. model_tier**

**Option A — as table**

```
CREATE TABLE model_tier (
  id SERIAL PRIMARY KEY,
  name VARCHAR(50) UNIQUE NOT NULL,    -- e.g. 'premium', 'standard', 'free'
  description TEXT
);
```

**Option B — as ENUM (cleaner if tiers don't change often)**

```
CREATE TYPE model_tier_enum AS ENUM ('premium', 'standard', 'free');
```

**5. model (reference tier)**

```
CREATE TABLE model (
  id SERIAL PRIMARY KEY,
  name VARCHAR(100) UNIQUE NOT NULL,   -- e.g. 'gpt-4o'
  tier_id INTEGER REFERENCES model_tier(id),
  created_at TIMESTAMP NOT NULL DEFAULT NOW()
);
```

**6. policy_scope enum**

```
CREATE TYPE policy_scope AS ENUM (
  'GLOBAL',
  'TENANT',
  'API_KEY',
  'MODEL',
  'MODEL_TIER',
  'USER_MODEL'
);
```

**7. rate_limit_policy**

```
CREATE TABLE rate_limit_policy (
  id SERIAL PRIMARY KEY,
  scope policy_scope NOT NULL,

  -- optional foreign keys depending on scope
  tenant_id      INTEGER REFERENCES tenant(id),
  user_id        INTEGER REFERENCES user_account(id),
  api_key_id     INTEGER REFERENCES api_key(id),
  model_id       INTEGER REFERENCES model(id),
  model_tier_id  INTEGER REFERENCES model_tier(id),
```

```
  window_seconds INTEGER NOT NULL,
  limit_value    INTEGER NOT NULL,

  enabled        BOOLEAN NOT NULL DEFAULT TRUE,
  created_at     TIMESTAMP NOT NULL DEFAULT NOW()
);
```