DATABASES SCRIPTS:

```
-- Table: public.articles

-- DROP TABLE IF EXISTS public.articles;

CREATE TABLE IF NOT EXISTS public.articles
(
    id uuid NOT NULL DEFAULT gen_random_uuid(),
    title character varying(255) COLLATE pg_catalog."default" NOT NULL,
    content text COLLATE pg_catalog."default" NOT NULL,
    author_id uuid,
    category character varying(255) COLLATE pg_catalog."default" NOT NULL,
    created_at timestamp(6) without time zone DEFAULT CURRENT_TIMESTAMP,
    updated_at timestamp(6) without time zone DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT articles_pkey PRIMARY KEY (id),
    CONSTRAINT articles_author_id_fkey FOREIGN KEY (author_id)
        REFERENCES public.users (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.articles
    OWNER to postgres;
-- Index: idx_articles_author_id

-- DROP INDEX IF EXISTS public.idx_articles_author_id;

CREATE INDEX IF NOT EXISTS idx_articles_author_id
    ON public.articles USING btree
    (author_id ASC NULLS LAST)
    TABLESPACE pg_default;

-- Table: public.cities

-- DROP TABLE IF EXISTS public.cities;

CREATE TABLE IF NOT EXISTS public.cities
(
    id uuid NOT NULL DEFAULT gen_random_uuid(),
    name character varying(255) COLLATE pg_catalog."default" NOT NULL,
    tax_level character varying(255) COLLATE pg_catalog."default" NOT NULL,
```

```
    climate_change_vulnerability character varying(255) COLLATE pg_catalog."default" NOT
NULL,
    politics character varying(255) COLLATE pg_catalog."default" NOT NULL,
    climate character varying(255) COLLATE pg_catalog."default" NOT NULL,
    average_income character varying(255) COLLATE pg_catalog."default" NOT NULL,
    natural_disasters character varying(255) COLLATE pg_catalog."default" NOT NULL,
    entertainment character varying(255) COLLATE pg_catalog."default" NOT NULL,
    average_age integer NOT NULL,
    upcoming boolean NOT NULL DEFAULT false,
    map_id uuid,
    CONSTRAINT cities_pkey PRIMARY KEY (id),
    CONSTRAINT cities_map_id_fkey FOREIGN KEY (map_id)
        REFERENCES public.maps (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.cities
    OWNER to postgres;
-- Index: idx_city_name

-- DROP INDEX IF EXISTS public.idx_city_name;

CREATE INDEX IF NOT EXISTS idx_city_name
    ON public.cities USING btree
    (name COLLATE pg_catalog."default" ASC NULLS LAST)
    TABLESPACE pg_default;
-- Index: idx_upcoming_cities

-- DROP INDEX IF EXISTS public.idx_upcoming_cities;

CREATE INDEX IF NOT EXISTS idx_upcoming_cities
    ON public.cities USING btree
    (upcoming ASC NULLS LAST)
    TABLESPACE pg_default;
-- Table: public.comments

-- DROP TABLE IF EXISTS public.comments;

CREATE TABLE IF NOT EXISTS public.comments
(
    id uuid NOT NULL DEFAULT gen_random_uuid(),
```

```
    article_id uuid NOT NULL,
    user_id uuid NOT NULL,
    content text COLLATE pg_catalog."default" NOT NULL,
    created_at timestamp(6) without time zone DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT comments_pkey PRIMARY KEY (id),
    CONSTRAINT comments_article_id_fkey FOREIGN KEY (article_id)
        REFERENCES public.articles (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT comments_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public.users (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.comments
    OWNER to postgres;
-- Index: idx_comments_article_id

-- DROP INDEX IF EXISTS public.idx_comments_article_id;

CREATE INDEX IF NOT EXISTS idx_comments_article_id
    ON public.comments USING btree
    (article_id ASC NULLS LAST)
    TABLESPACE pg_default;

-- Table: public.favorites

-- DROP TABLE IF EXISTS public.favorites;

CREATE TABLE IF NOT EXISTS public.favorites
(
    id uuid NOT NULL DEFAULT gen_random_uuid(),
    article_id uuid NOT NULL,
    user_id uuid NOT NULL,
    created_at timestamp(6) without time zone DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT favorites_pkey PRIMARY KEY (id),
    CONSTRAINT favorites_article_id_fkey FOREIGN KEY (article_id)
        REFERENCES public.articles (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT favorites_user_id_fkey FOREIGN KEY (user_id)
```

```
        REFERENCES public.users (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.favorites
    OWNER to postgres;
-- Index: idx_favorites_article_id

-- DROP INDEX IF EXISTS public.idx_favorites_article_id;

CREATE INDEX IF NOT EXISTS idx_favorites_article_id
    ON public.favorites USING btree
    (article_id ASC NULLS LAST)
    TABLESPACE pg_default;
-- Table: public.maps

-- DROP TABLE IF EXISTS public.maps;

CREATE TABLE IF NOT EXISTS public.maps
(
    id uuid NOT NULL DEFAULT gen_random_uuid(),
    name character varying(255) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT maps_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.maps
    OWNER to postgres;
-- Table: public.users

-- DROP TABLE IF EXISTS public.users;

CREATE TABLE IF NOT EXISTS public.users
(
    id uuid NOT NULL DEFAULT gen_random_uuid(),
    username character varying(255) COLLATE pg_catalog."default" NOT NULL,
    email character varying(255) COLLATE pg_catalog."default" NOT NULL,
    password character varying(255) COLLATE pg_catalog."default" NOT NULL,
    role character varying(255) COLLATE pg_catalog."default" DEFAULT 'user'::text,
    created_at timestamp(6) without time zone DEFAULT CURRENT_TIMESTAMP,
```

```
    full_name character varying(255) COLLATE pg_catalog."default",
    enabled boolean NOT NULL DEFAULT true,
    CONSTRAINT users_pkey PRIMARY KEY (id),
    CONSTRAINT users_email_key UNIQUE (email),
    CONSTRAINT users_username_key UNIQUE (username),
    CONSTRAINT users_role_check CHECK (role::text = ANY (ARRAY['user'::text,
'admin'::text]))
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.users
    OWNER to postgres;
-- Table: public.votes

-- DROP TABLE IF EXISTS public.votes;

CREATE TABLE IF NOT EXISTS public.votes
(
    id uuid NOT NULL DEFAULT gen_random_uuid(),
    target_id uuid NOT NULL,
    target_type character varying(255) COLLATE pg_catalog."default" NOT NULL,
    user_id uuid NOT NULL,
    vote_type character varying(255) COLLATE pg_catalog."default" NOT NULL,
    created_at timestamp(6) without time zone DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT votes_pkey PRIMARY KEY (id),
    CONSTRAINT votes_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public.users (id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT votes_target_type_check CHECK (target_type::text = ANY
(ARRAY['post'::text, 'article'::text, 'comment'::text])),
    CONSTRAINT votes_type_check CHECK (vote_type::text = ANY (ARRAY['upvote'::text,
'downvote'::text]))
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.votes
    OWNER to postgres;
```