**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ ANABİLİM DALI**

**BİLİŞİM TEKNOLOJİLERİ PR. (YL) (UZAKTAN EĞİTİM)**



EBT514- VERİTABANI TASARIMI VE YÖNETİMİ DERSİ ÖDEVİ

e225013176

Zeliha MUTLU ÇAKIN

zeliha.cakin@ogr.sakarya.edu.tr

Dr.Öğr.Üyesi İSMAİL ÖZTEL

*TANIM:*

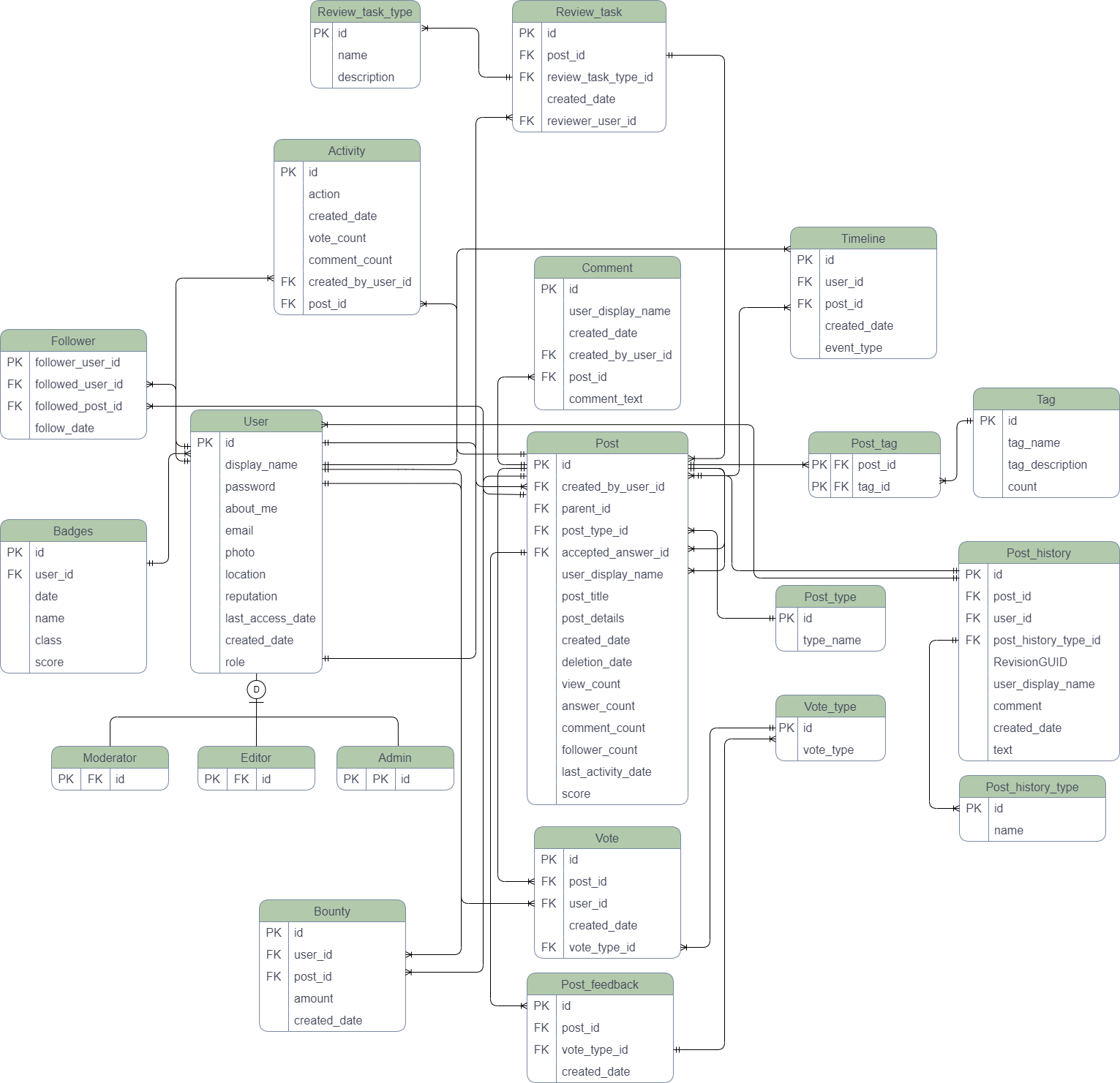
Amacı belli projeleri amacına ulaştırmaya çalışan yazılım geliştiriciler programlama faliyetleri sırasında yeni problem ve sorularla karşılaşabilmektedirler. Teknolojinin sürekli değişmesi ve yenilenmesiyle beraber yazılım geliştiriciler de sürekli öğrenmek ve yeni sorunlara çözüm üretmek durumunda kalmaktadır. Bu süreçte, uzun dokümantasyonlar okumak yerine, nokta atışı sorular ve yanıtlar, yazılım geliştiricilere problem çözme noktasında zaman kazandıran bir yöntem olarak öne çıkmaktadır.

Buna bir alternatif çözüm yolu olarak tüm dünya çapında bilişim sektöründeki insanları ortak bir platformda buluşturarak ve sorunlarına ortak ve en doğru çözümü bulmak amacıyla bir soru cevap(Q&A) platformu oluşturulacak ve kullanıcılar bu sayede birbiriyle bilgi paylaşımı yapabilecektir.

Geliştirilecek yazılımda; kayıtlı kullanıcıllar ve aynı zamanda misafir kullanıcılar bulunacaktır.Kullanıcılar soru ve cevaplarından geri bildirimlerle puan kazanacak, sorulara ve yanıtlara oy verebilecek ve bu şekilde kaliteli içeriği vurgulayabilecektir.Kullanıcılar sorularını uygun kategorilere etiketleyebilecek ve soruların doğru hedef kitleye ulaşmasını ve ilgili uzmanların dikkatini çekmesini sağlayabilecektir.

*İŞ KURALLARI:*

1. Site kullanıcıları moderatörler, editörler, yöneticiler ve kayıtlı kullanıcılardan oluşur.
2. Siteyi misafir ve üye kullanıcılar ziyaret edebilecektir. Misafir kullanıcılar ise kayıt oluşturabilir, üye olabilir.
3. Kaydı yapılan her bir üyenin ad, soyad, şifre, fotoğraf ,e-mail, adres , hesap oluşturma tarihi, son giriş tarihi, kendi hakkında yazısı olacaktır.
4. Kullanıcılar birbirinden ID ile ayırt edilebilecektir.
5. Her kullanıcı birden çok soru sorabilir, birden çok soru yanıtlayabilir, birden çok makalesi olabilir. Ancak bir sorunun, makalenin veya yanıtın oluşturan kullanıcısı bir tanedir.
6. Kullanıcılar birden çok kullanıcıyı ve birden çok gönderiyi takip edebilir.Takibe başlanan tarih tutulur.
7. Her kullanıcı birçok gönderiye oy kullanabilir ancak kullanıcının bir gönderiye bir oy kullanma hakkı vardır.Oylar ulumlu yada olumsuz olabilir.Oy kullanma tarihi tutulur.
8. Her kullanıcı rozet ile derecelendirilecektir. Her rozetin adı puan bilgisi, kazanıldığı tarih ve türü bilgisi tutulur.Rozetler puanlarına göre derecelendirilir.Bir kullanıcının birden çok rozeti olabilir, bir rozet bir kullanıcıya aittir.
9. Gönderiler soru, cevap, makale şeklinde olabilir.
10. Her gönderinin bir başlığı, açıklaması, oluşturulma tarihi, kim tarafında oluşturulduğu bilgisi , görüntülenme sayısı, cevap sayısı, cevaplayan kullanıcı bilgisi, oy sayısı, etiketi oluşturulma tarihi bulunur. Timeline düzenlenir.
11. Her cevabın metni ne zaman oluşturulduğu,yazan kullanıcı bilgisi vardır.
12. Gönderilerinden ödül kazanabilir.Ödülün rütbesi ve kazanıldığı tarih bilgisi tutulur.bir kullanıcı birden fazla ödül kazanabilir ve birden fazla gönderiden ödül kazanılabilir.
13. Sorulan her bir soruda birden fazla etiket kullanılarak kategorilerde ön plana çıkarılır.Bir etiket bir çok gönderide kullanılabilir ve bir gönderinin de bir çok etiketi olabilir.
14. Her gönderinin tarihsel kaydı tutulur.
15. Her kullanıcının soru cevap ve oy verme gibi aktivitelerini tutan bir aktivite kaydı vardır.
16. Kullanıcı birçok gönderiye yorum yapabilir ancak bir yorumun bir oluşturan kullanıcısı vardır. Yorumun metni, oluşturulma tarihi, oluşturan kullanıcı bilgisi, görüntüleme ve oylama bilgisi vardır.bir gönderiye birden fazla yorum yapılabilir.
17. Moderatörler , yöneticiler ve editörle gönderileri inceleme, kapatma gibi görevlere sahiptir.



Varlık Bağıntı Diyagramı(Crow’s Foot)

*İLİŞKİSEL ŞEMA (Metinsel Gösterim):*

Post(**id:integer**, created\_by\_user\_id:integer, parent\_id:integer, accepted\_answer\_id:integer, post\_type\_id:smallint, user\_display\_name: character varying (50), post\_title:character varying(250), post\_details:character varying, created\_date:timestamp(4) without time zone, deletion\_date:timestamp(4) without time zone, score:integer, view\_count:integer, answer\_count:integer, comment\_count:integer, follower\_count:integer, last\_activity\_date:timestamp(4) without time zone)

Post\_feedback(**id:integer**, post\_id:integer, vote\_type\_id:smallint, creatde\_date:timestamp(4) without time zone)

Post\_history(**id:integer**, post\_history\_type\_id:smallint, post\_id:integer, created\_date:timestamp(4) without time zone, user\_id:integer, user\_display\_name:character varying(50), comment:character varying(400), text:character varying, RevisionGUID:integer)

Post\_history\_type(**id:smallint**, name:character varying(50))

Post\_tag(**tag\_id:integer, post\_id:integer**)

Post\_type(**id:smallint**, type\_name:character varying(50))

Review\_task(**id:integer**, review\_task\_type\_id:smallint, created\_date:timestamp(4) without time zone, post\_id:integer, reviewer\_user\_id:integer)

Review\_task\_type(**id smallint**, name:character varying(50), description: character varying(300))

Tag(**id:integer**, count:integer, tag\_name:character varying(40),tag\_description:character varying(250))

Activity(**id:integer**, vote\_count:integer, post\_id:integer, created\_date:timestamp without time zone, comment\_count:integer, created\_by\_user\_id: integer, actions: character varying (255) )

Badges(**id:integer**, user\_id:integer, name: character varying (50), date:timestamp(4) without time zone, class:smallint, score:integer)

Bounty(**id:integer**, post\_id:integer, user\_id:integer, amount:integer, created\_date: timestamp (4) without time zone)

Comment(**id:integer**, post\_id:integer, comment\_text: character varying (600), created\_date:timestamp(4) without time zone, user\_display\_name: character varying (50), created\_by\_user\_id:integer)

Follower(**follower\_id:integer**, followed\_user\_id:integer, followed\_post\_id :integer, follow\_date:timestamp(4) without time zone)

Timeline(**id integer**, user\_id:integer, post\_id:integer, created\_date:timestamp without time zone, event\_type:character varying(255))

User(**id:integer**, display\_name:character varying(50), reputation:integer, about\_me:character varying, email:character varying(32), created\_date:timestamp(4) without time zone, photo:character varying(200), location:character varying(100), last\_access\_date:timestamp(4) without time zone, password:character varying(100), role:character varying(20))

Vote(**id:integer**, post\_id:integer, vote\_type\_id:smallint, user\_id:integer, created\_date:timestamp(4) without time zone)

Vote\_type(**id:smallint**, vote\_type:character varying(50))

Moderator(**id:integer**)

Editor(**id:integer**)

Admin(**id:integer**)

*TASARLANMIŞ OLAN Q&A DATABASE SQL KODLARI:*

--

-- PostgreSQL database dump

--

-- Dumped from database version 15.3

-- Dumped by pg\_dump version 15rc2

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

--

-- Name: User; Type: SCHEMA; Schema: -; Owner: postgres

--

CREATE SCHEMA "User";

ALTER SCHEMA "User" OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

-- Name: Admin; Type: TABLE; Schema: User; Owner: postgres

--

CREATE TABLE "User"."Admin" (

id integer NOT NULL

);

ALTER TABLE "User"."Admin" OWNER TO postgres;

--

-- Name: Editor; Type: TABLE; Schema: User; Owner: postgres

--

CREATE TABLE "User"."Editor" (

id integer NOT NULL

);

ALTER TABLE "User"."Editor" OWNER TO postgres;

--

-- Name: Moderator; Type: TABLE; Schema: User; Owner: postgres

--

CREATE TABLE "User"."Moderator" (

id integer NOT NULL

);

ALTER TABLE "User"."Moderator" OWNER TO postgres;

--

-- Name: Activity; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Activity" (

vote\_count integer,

post\_id integer NOT NULL,

created\_date timestamp without time zone NOT NULL,

id integer NOT NULL,

comment\_count integer,

created\_by\_user\_id integer NOT NULL,

actions character varying DEFAULT '255'::character varying

);

ALTER TABLE public."Activity" OWNER TO postgres;

--

-- Name: Badges; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Badges" (

id integer NOT NULL,

user\_id integer,

name character varying(50),

date timestamp(4) without time zone,

class smallint,

score integer

);

ALTER TABLE public."Badges" OWNER TO postgres;

--

-- Name: Bounty; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Bounty" (

id integer NOT NULL,

post\_id integer,

user\_id integer,

amount integer,

created\_date timestamp(4) without time zone

);

ALTER TABLE public."Bounty" OWNER TO postgres;

--

-- Name: Comment; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Comment" (

id integer NOT NULL,

post\_id integer,

comment\_text character varying(600),

created\_date timestamp(4) without time zone,

user\_display\_name character varying(50),

created\_by\_user\_id integer

);

ALTER TABLE public."Comment" OWNER TO postgres;

--

-- Name: Follower; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Follower" (

follower\_id integer NOT NULL,

followed\_user\_id integer NOT NULL,

followed\_post\_id integer NOT NULL,

follow\_date timestamp(4) without time zone

);

ALTER TABLE public."Follower" OWNER TO postgres;

--

-- Name: Post; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Post" (

id integer NOT NULL,

created\_by\_user\_id integer,

parent\_id integer,

accepted\_answer\_id integer,

post\_type\_id smallint,

user\_display\_name character varying(50),

post\_title character varying(250),

post\_details character varying,

created\_date timestamp(4) without time zone,

deletion\_date timestamp(4) without time zone,

score integer,

view\_count integer,

answer\_count integer,

comment\_count integer,

follower\_count integer,

last\_activity\_date timestamp(4) without time zone

);

ALTER TABLE public."Post" OWNER TO postgres;

--

-- Name: Post\_feedback; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Post\_feedback" (

id integer NOT NULL,

post\_id integer,

vote\_type\_id smallint,

creatde\_date timestamp(4) without time zone

);

ALTER TABLE public."Post\_feedback" OWNER TO postgres;

--

-- Name: Post\_history; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Post\_history" (

id integer NOT NULL,

post\_history\_type\_id smallint,

post\_id integer,

created\_date timestamp(4) without time zone,

user\_id integer,

user\_display\_name character varying(50),

comment character varying(400),

text character varying,

"RevisionGUID" integer NOT NULL

);

ALTER TABLE public."Post\_history" OWNER TO postgres;

--

-- Name: Post\_history\_RevisionGUID\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

CREATE SEQUENCE public."Post\_history\_RevisionGUID\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER TABLE public."Post\_history\_RevisionGUID\_seq" OWNER TO postgres;

--

-- Name: Post\_history\_RevisionGUID\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres

--

ALTER SEQUENCE public."Post\_history\_RevisionGUID\_seq" OWNED BY public."Post\_history"."RevisionGUID";

--

-- Name: Post\_history\_type; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Post\_history\_type" (

id smallint NOT NULL,

name character varying(50)

);

ALTER TABLE public."Post\_history\_type" OWNER TO postgres;

--

-- Name: Post\_tag; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Post\_tag" (

tag\_id integer NOT NULL,

post\_id integer NOT NULL

);

ALTER TABLE public."Post\_tag" OWNER TO postgres;

--

-- Name: Post\_type; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Post\_type" (

id smallint NOT NULL,

type\_name character varying(50)

);

ALTER TABLE public."Post\_type" OWNER TO postgres;

--

-- Name: Review\_task; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Review\_task" (

id integer NOT NULL,

review\_task\_type\_id smallint,

created\_date timestamp(4) without time zone,

post\_id integer,

reviewer\_user\_id integer

);

ALTER TABLE public."Review\_task" OWNER TO postgres;

--

-- Name: Review\_task\_type; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Review\_task\_type" (

id smallint NOT NULL,

name character varying(50),

description character varying(300)

);

ALTER TABLE public."Review\_task\_type" OWNER TO postgres;

--

-- Name: Tag; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Tag" (

id integer NOT NULL,

count integer,

tag\_name character varying(40),

tag\_description character varying(250)

);

ALTER TABLE public."Tag" OWNER TO postgres;

--

-- Name: Timeline; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Timeline" (

id integer NOT NULL,

user\_id integer NOT NULL,

post\_id integer NOT NULL,

created\_date timestamp without time zone NOT NULL,

event\_type character varying DEFAULT '255'::character varying NOT NULL

);

ALTER TABLE public."Timeline" OWNER TO postgres;

--

-- Name: User; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."User" (

id integer NOT NULL,

display\_name character varying(50),

reputation integer,

about\_me character varying,

email character varying(32),

created\_date timestamp(4) without time zone,

photo character varying(200),

location character varying(100),

last\_access\_date timestamp(4) without time zone,

password character varying(100),

role character varying(20)

);

ALTER TABLE public."User" OWNER TO postgres;

--

-- Name: Vote; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Vote" (

id integer NOT NULL,

post\_id integer,

"Vote\_type\_id" smallint,

user\_id integer,

created\_date timestamp(4) without time zone

);

ALTER TABLE public."Vote" OWNER TO postgres;

--

-- Name: Vote\_type; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Vote\_type" (

id smallint NOT NULL,

vote\_type character varying(50)

);

ALTER TABLE public."Vote\_type" OWNER TO postgres;

--

-- Data for Name: Admin; Type: TABLE DATA; Schema: User; Owner: postgres

--

--

-- Data for Name: Editor; Type: TABLE DATA; Schema: User; Owner: postgres

--

--

-- Data for Name: Moderator; Type: TABLE DATA; Schema: User; Owner: postgres

--

--

-- Data for Name: Activity; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Badges; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Bounty; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Comment; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Follower; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Post; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Post\_feedback; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Post\_history; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Post\_history\_type; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Post\_tag; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Post\_type; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Review\_task; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Review\_task\_type; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Tag; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Timeline; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: User; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Vote; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: Vote\_type; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Name: Post\_history\_RevisionGUID\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

SELECT pg\_catalog.setval('public."Post\_history\_RevisionGUID\_seq"', 1, false);

--

-- Name: Admin AdminPK; Type: CONSTRAINT; Schema: User; Owner: postgres

--

ALTER TABLE ONLY "User"."Admin"

ADD CONSTRAINT "AdminPK" PRIMARY KEY (id);

--

-- Name: Editor EditorPK; Type: CONSTRAINT; Schema: User; Owner: postgres

--

ALTER TABLE ONLY "User"."Editor"

ADD CONSTRAINT "EditorPK" PRIMARY KEY (id);

--

-- Name: Moderator ModeratorPK; Type: CONSTRAINT; Schema: User; Owner: postgres

--

ALTER TABLE ONLY "User"."Moderator"

ADD CONSTRAINT "ModeratorPK" PRIMARY KEY (id);

--

-- Name: Activity Activity\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Activity"

ADD CONSTRAINT "Activity\_pkey" PRIMARY KEY (id);

--

-- Name: Badges BadgesPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Badges"

ADD CONSTRAINT "BadgesPK" PRIMARY KEY (id);

--

-- Name: Bounty BountyPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bounty"

ADD CONSTRAINT "BountyPK" PRIMARY KEY (id);

--

-- Name: Comment CommentPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Comment"

ADD CONSTRAINT "CommentPK" PRIMARY KEY (id);

--

-- Name: Follower Follower\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Follower"

ADD CONSTRAINT "Follower\_pkey" PRIMARY KEY (follower\_id);

--

-- Name: Post PostPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post"

ADD CONSTRAINT "PostPK" PRIMARY KEY (id);

--

-- Name: Post\_feedback Post\_feedbackPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_feedback"

ADD CONSTRAINT "Post\_feedbackPK" PRIMARY KEY (id);

--

-- Name: Post\_history Post\_historyPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_history"

ADD CONSTRAINT "Post\_historyPK" PRIMARY KEY (id);

--

-- Name: Post\_history\_type Post\_history\_typePK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_history\_type"

ADD CONSTRAINT "Post\_history\_typePK" PRIMARY KEY (id);

--

-- Name: Post\_tag Post\_tag\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_tag"

ADD CONSTRAINT "Post\_tag\_pkey" PRIMARY KEY (post\_id, tag\_id);

--

-- Name: Post\_type Post\_typePK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_type"

ADD CONSTRAINT "Post\_typePK" PRIMARY KEY (id);

--

-- Name: Review\_task Review\_taskPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Review\_task"

ADD CONSTRAINT "Review\_taskPK" PRIMARY KEY (id);

--

-- Name: Review\_task\_type Review\_task\_typePK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Review\_task\_type"

ADD CONSTRAINT "Review\_task\_typePK" PRIMARY KEY (id);

--

-- Name: Tag Tag\_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Tag"

ADD CONSTRAINT "Tag\_pkey" PRIMARY KEY (id);

--

-- Name: User UserPK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."User"

ADD CONSTRAINT "UserPK" PRIMARY KEY (id);

--

-- Name: Vote VotePK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Vote"

ADD CONSTRAINT "VotePK" PRIMARY KEY (id);

--

-- Name: Vote\_type Vote\_typePK; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Vote\_type"

ADD CONSTRAINT "Vote\_typePK" PRIMARY KEY (id);

--

-- Name: Activity unique\_Activity\_created\_by\_user\_id; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Activity"

ADD CONSTRAINT "unique\_Activity\_created\_by\_user\_id" UNIQUE (created\_by\_user\_id);

--

-- Name: Activity unique\_Activity\_post\_id; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Activity"

ADD CONSTRAINT "unique\_Activity\_post\_id" UNIQUE (post\_id);

--

-- Name: Timeline unique\_Timeline\_id; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Timeline"

ADD CONSTRAINT "unique\_Timeline\_id" PRIMARY KEY (id);

--

-- Name: Admin AdminUser; Type: FK CONSTRAINT; Schema: User; Owner: postgres

--

ALTER TABLE ONLY "User"."Admin"

ADD CONSTRAINT "AdminUser" FOREIGN KEY (id) REFERENCES "User"."Admin"(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Editor EditorUser; Type: FK CONSTRAINT; Schema: User; Owner: postgres

--

ALTER TABLE ONLY "User"."Editor"

ADD CONSTRAINT "EditorUser" FOREIGN KEY (id) REFERENCES "User"."Editor"(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Moderator ModeratorUser; Type: FK CONSTRAINT; Schema: User; Owner: postgres

--

ALTER TABLE ONLY "User"."Moderator"

ADD CONSTRAINT "ModeratorUser" FOREIGN KEY (id) REFERENCES "User"."Moderator"(id) ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Activity ActivityFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Activity"

ADD CONSTRAINT "ActivityFK1" FOREIGN KEY (post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Activity ActivityFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Activity"

ADD CONSTRAINT "ActivityFK2" FOREIGN KEY (created\_by\_user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Badges BadgesFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Badges"

ADD CONSTRAINT "BadgesFK1" FOREIGN KEY (user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Bounty BountyFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bounty"

ADD CONSTRAINT "BountyFK1" FOREIGN KEY (user\_id) REFERENCES public."User"(id);

--

-- Name: Bounty BountyFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bounty"

ADD CONSTRAINT "BountyFK2" FOREIGN KEY (post\_id) REFERENCES public."Post"(id);

--

-- Name: Comment CommentFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Comment"

ADD CONSTRAINT "CommentFK1" FOREIGN KEY (created\_by\_user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Comment CommentFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Comment"

ADD CONSTRAINT "CommentFK2" FOREIGN KEY (post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Follower FollowerFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Follower"

ADD CONSTRAINT "FollowerFK1" FOREIGN KEY (followed\_post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Follower FollowerFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Follower"

ADD CONSTRAINT "FollowerFK2" FOREIGN KEY (followed\_user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post PostFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post"

ADD CONSTRAINT "PostFK1" FOREIGN KEY (created\_by\_user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post PostFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post"

ADD CONSTRAINT "PostFK2" FOREIGN KEY (parent\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post PostFK3; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post"

ADD CONSTRAINT "PostFK3" FOREIGN KEY (post\_type\_id) REFERENCES public."Post\_type"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post PostFK4; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post"

ADD CONSTRAINT "PostFK4" FOREIGN KEY (accepted\_answer\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post\_feedback Post\_feedbackFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_feedback"

ADD CONSTRAINT "Post\_feedbackFK1" FOREIGN KEY (post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post\_feedback Post\_feedbackFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_feedback"

ADD CONSTRAINT "Post\_feedbackFK2" FOREIGN KEY (vote\_type\_id) REFERENCES public."Vote\_type"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post\_history Post\_historyFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_history"

ADD CONSTRAINT "Post\_historyFK1" FOREIGN KEY (post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post\_history Post\_historyFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_history"

ADD CONSTRAINT "Post\_historyFK2" FOREIGN KEY (user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Post\_history Post\_historyFK3; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Post\_history"

ADD CONSTRAINT "Post\_historyFK3" FOREIGN KEY (post\_history\_type\_id) REFERENCES public."Post\_history\_type"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Review\_task Review\_taskFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Review\_task"

ADD CONSTRAINT "Review\_taskFK1" FOREIGN KEY (review\_task\_type\_id) REFERENCES public."Review\_task\_type"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Review\_task Review\_taskFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Review\_task"

ADD CONSTRAINT "Review\_taskFK2" FOREIGN KEY (post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Review\_task Review\_taskFK3; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Review\_task"

ADD CONSTRAINT "Review\_taskFK3" FOREIGN KEY (reviewer\_user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Timeline TimelineFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Timeline"

ADD CONSTRAINT "TimelineFK1" FOREIGN KEY (post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Timeline TimelineFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Timeline"

ADD CONSTRAINT "TimelineFK2" FOREIGN KEY (user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Vote VoteFK1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Vote"

ADD CONSTRAINT "VoteFK1" FOREIGN KEY (post\_id) REFERENCES public."Post"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Vote VoteFK2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Vote"

ADD CONSTRAINT "VoteFK2" FOREIGN KEY (user\_id) REFERENCES public."User"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: Vote VoteFK3; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Vote"

ADD CONSTRAINT "VoteFK3" FOREIGN KEY ("Vote\_type\_id") REFERENCES public."Vote\_type"(id) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- PostgreSQL database dump complete

--

);