Database schema:

1. Answers:

CREATE TABLE public.answers

(

id bigint NOT NULL,

description character varying(255) COLLATE pg\_catalog."default",

downvote\_count bigint NOT NULL,

upvote\_count bigint NOT NULL,

owner\_question\_id bigint,

owner\_user\_id bigint,

session\_id bigint,

CONSTRAINT answers\_pkey PRIMARY KEY (id),

CONSTRAINT fk1clk1fob80d1ok7hm2dbd1tcr FOREIGN KEY (owner\_question\_id)

REFERENCES public.questions (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE,

CONSTRAINT fk7syc0on1l11b4e9oay2c0qf8a FOREIGN KEY (owner\_user\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE,

CONSTRAINT fkqjcmodo262qk6jk2rym73v0nj FOREIGN KEY (session\_id)

REFERENCES public.discussion\_session (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

);

2. Discussion Session:

CREATE TABLE public.discussion\_session

(

id bigint NOT NULL DEFAULT nextval('discussion\_session\_id\_seq'::regclass),

discussiontime timestamp without time zone,

status character varying(255) COLLATE pg\_catalog."default",

expert\_id bigint,

question\_id bigint,

seeker\_id bigint,

CONSTRAINT discussion\_session\_pkey PRIMARY KEY (id),

CONSTRAINT fk2r9jd14urtlmcigp0u1nftqjs FOREIGN KEY (expert\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE,

CONSTRAINT fk8hbojirm59ln1f3n0s6listo7 FOREIGN KEY (question\_id)

REFERENCES public.questions (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE,

CONSTRAINT fkh3a264igi2emk5xfl2bc1wqf5 FOREIGN KEY (seeker\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

);

3. Users:

CREATE TABLE public.final\_ucan\_users

(

id bigint NOT NULL DEFAULT nextval('final\_ucan\_users\_id\_seq'::regclass),

communication\_lang character varying(255) COLLATE pg\_catalog."default",

communication\_mode character varying(255) COLLATE pg\_catalog."default",

communication\_time character varying(255) COLLATE pg\_catalog."default",

contact character varying(255) COLLATE pg\_catalog."default",

country character varying(255) COLLATE pg\_catalog."default",

email character varying(255) COLLATE pg\_catalog."default",

name character varying(255) COLLATE pg\_catalog."default",

password character varying(255) COLLATE pg\_catalog."default",

user\_name character varying(255) COLLATE pg\_catalog."default",

CONSTRAINT final\_ucan\_users\_pkey PRIMARY KEY (id)

);

4. Topics:

CREATE TABLE public.topic

(

id bigint NOT NULL,

follower\_count character varying(255) COLLATE pg\_catalog."default",

name character varying(255) COLLATE pg\_catalog."default",

CONSTRAINT topic\_pkey PRIMARY KEY (id)

);

5. Questions:

CREATE TABLE public.questions

(

id bigint NOT NULL,

communication\_mode character varying(255) COLLATE pg\_catalog."default",

preferred\_language character varying(255) COLLATE pg\_catalog."default",

preferred\_time character varying(255) COLLATE pg\_catalog."default",

question\_text character varying(255) COLLATE pg\_catalog."default",

title character varying(255) COLLATE pg\_catalog."default",

owner\_user\_id bigint,

downvote\_count bigint NOT NULL,

upvote\_count bigint NOT NULL,

CONSTRAINT questions\_pkey PRIMARY KEY (id),

CONSTRAINT fkqu2cfddw6w56lruw5bo5bg1v0 FOREIGN KEY (owner\_user\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

);

6. Question tags:

CREATE TABLE public.questions\_tags

(

question\_id bigint NOT NULL,

tags\_id bigint NOT NULL,

CONSTRAINT fk4u5xv906wfevngoe973bec6u0 FOREIGN KEY (question\_id)

REFERENCES public.questions (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fkgj9mse84ikryibaaxsetvyrf4 FOREIGN KEY (tags\_id)

REFERENCES public.topic (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

);

7. Negotiation Session

CREATE TABLE public.negotiation\_session

(

id bigint NOT NULL DEFAULT nextval('negotiation\_session\_id\_seq'::regclass),

status character varying(255) COLLATE pg\_catalog."default",

expert\_id bigint,

seeker\_id bigint,

CONSTRAINT negotiation\_session\_pkey PRIMARY KEY (id),

CONSTRAINT fkcgip5erlaxm9ywch5x14rx1fk FOREIGN KEY (expert\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE,

CONSTRAINT fkqci01vjg7m07254vlvmabsit2 FOREIGN KEY (seeker\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

);

8. Negotiation Messages:

CREATE TABLE public.negotiation\_messages

(

id bigint NOT NULL,

message\_text character varying(255) COLLATE pg\_catalog."default",

nego\_session\_id bigint,

owner\_user\_id bigint,

CONSTRAINT negotiation\_messages\_pkey PRIMARY KEY (id),

CONSTRAINT fkije2p6270rya1948torhqdysk FOREIGN KEY (nego\_session\_id)

REFERENCES public.negotiation\_session (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE,

CONSTRAINT fkjqcuy3i8afqxoknvh5w3ef1p1 FOREIGN KEY (owner\_user\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

);

9.User Expertise topics:

CREATE TABLE public.final\_ucan\_users\_expertise\_topics

(

user\_id bigint NOT NULL,

expertise\_topics\_id bigint NOT NULL,

CONSTRAINT final\_ucan\_users\_expertise\_topics\_pkey PRIMARY KEY (user\_id, expertise\_topics\_id),

CONSTRAINT fk2bn3uqfrd2c93jji08qwx83wy FOREIGN KEY (user\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fkeapalrwi5bkkbc1936u2wkkex FOREIGN KEY (expertise\_topics\_id)

REFERENCES public.topic (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

);

10. User Interested topics:

CREATE TABLE public.final\_ucan\_users\_interested\_topics

(

user\_id bigint NOT NULL,

interested\_topics\_id bigint NOT NULL,

CONSTRAINT fkj5yt68j3t09nxapnjd9rg246v FOREIGN KEY (interested\_topics\_id)

REFERENCES public.topic (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fkr8jys0sx5nq4bp0sv4ecg6cxo FOREIGN KEY (user\_id)

REFERENCES public.final\_ucan\_users (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

);