Schema:

Users

user_id(pk)
name
passwd
user_type(student,supervisor,reviewer,deanap)
token (stored for forget password)

Student

student_id (fk - references user_id in users table)
research_area
supervisor_id (fk -references user_id in students table)
synopsis (synopsis pdf stored as bytea type)
synopsis_date
thesis_id (fk - references thesis_id in thesis table)

Thesis

thesis_id(pk)
thesis_data (thesis pdf stored as bytea type)
thesis_name
submitted_date
status (submitted, accepted,reviewersSelected,Reviewing, Reviewed,defended)

Reviewer

email_id(pk,fk-references user_id in users table)
name
affiliation
designation
contact_no
address
type
supervisor_id (fk - references user_id in users table)

Review

thesis_id(pk)
email_id(pk (Email ID of reviewer), fk-references user_id in users table)
review (review pdf stored as bytea type)

ThesisReviewer: shows the relation between thesis , reviewer and supervisor

thesis_id(pk,fk-references thesis_id in thesis table)
reviewer_id(pk,fk-references user_id in users table)
supervisor_id(pk, fk-references user_id in users table)
Token (token for accepting thesis,used when invitation mail is sent)
sentdate (date when the invitation mail is sent)
status(Selected, reviewerAccepted, deanAccepted, addedToDashboard, reviewSent,sentToSupervisor)
accepteddate(date when the reviewer has accepted the thesis)

Dean Account details:stores the email,password of dean through which invitation,notification mail is sent

Email(pk)
Password
invitation_mail_body (default invitation mail body)
invitation_mail_subject(default invitation mail subject)
notification_mail_body(default notification mail body)
notification_mail_subject(default notification mail subject)

Tempstudent:stores the student details temporarily for signup until the student verfies the email

email(pk)
password
supervisor_id(fk-references user_id in users table)
research_area
name
Token (token for verifying email)

DDL Scripts:

1. **Users table :**

-- Table: public.users

```
-- DROP TABLE public.users;

CREATE TABLE public.users
(
    user_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    user_name character varying(100) COLLATE pg_catalog."default" NOT NULL,
    passwd character varying(100) COLLATE pg_catalog."default" NOT NULL,
    user_type character varying(100) COLLATE pg_catalog."default" NOT NULL,
    token character varying COLLATE pg_catalog."default",
    CONSTRAINT users_pkey PRIMARY KEY (user_id)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

```
ALTER TABLE public.users
    OWNER to postgres;
```

2.Student table

```
-- Table: public.student

-- DROP TABLE public.student;

CREATE TABLE public.student
(
    student_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    research_area character varying(100) COLLATE pg_catalog."default" NOT NULL,
    supervisor_id character varying(100) COLLATE pg_catalog."default",
    synopsis_date date,
    thesis_id integer,
    synopsis bytea,
    CONSTRAINT student_pkey PRIMARY KEY (student_id),
    CONSTRAINT student_student_id_fkey FOREIGN KEY (student_id)
        REFERENCES public.users (user_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT student_supervisor_id_fkey FOREIGN KEY (supervisor_id)
        REFERENCES public.users (user_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE SET NULL,
    CONSTRAINT student_thesis_id_fkey FOREIGN KEY (thesis_id)
        REFERENCES public.thesis (thesis_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE SET NULL
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.student
    OWNER to postgres;
```

3. **Thesis table:**

```
-- Table: public.thesis

-- DROP TABLE public.thesis;

CREATE TABLE public.thesis
(
    thesis_id integer NOT NULL DEFAULT nextval('thesis_thesis_id_seq'::regclass),
    thesis_name character varying(100) COLLATE pg_catalog."default" NOT NULL,
    submitted_date date,
    status character varying(100) COLLATE pg_catalog."default",
    thesis_data bytea,
    CONSTRAINT thesis_pkey PRIMARY KEY (thesis_id)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.thesis
    OWNER to postgres;
```

4. **Reviewer table:**

```
-- Table: public.reviewer

-- DROP TABLE public.reviewer;

CREATE TABLE public.reviewer
(
    email_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    affiliation character varying(100) COLLATE pg_catalog."default" NOT NULL,
    designation character varying(100) COLLATE pg_catalog."default",
    address character varying(500) COLLATE pg_catalog."default",
    reviewer_type character varying(100) COLLATE pg_catalog."default" NOT NULL,
    contact_no character varying(20) COLLATE pg_catalog."default",
    supervisor_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    name character varying(100) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT reviewer_pkey PRIMARY KEY (email_id, supervisor_id),
    CONSTRAINT reviewer_supervisor_id_fkey FOREIGN KEY (supervisor_id)
        REFERENCES public.users (user_id) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE CASCADE
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.reviewer
    OWNER to postgres;
```

5. **Review table:**

```
-- Table: public.review

-- DROP TABLE public.review;

CREATE TABLE public.review
(
    thesis_id integer NOT NULL,
    email_id character varying(100) COLLATE pg_catalog."default" NOT NULL,
    review bytea NOT NULL,
    submissiondate date NOT NULL,
    CONSTRAINT review_pkey PRIMARY KEY (thesis_id, email_id),
    CONSTRAINT review_email_id_fkey FOREIGN KEY (email_id)
        REFERENCES public.users (user_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT review_thesis_id_fkey FOREIGN KEY (thesis_id)
        REFERENCES public.thesis (thesis_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.review
    OWNER to postgres;
```

6. **deanaccountdetails table:**

```
-- Table: public.deanaccountdetails

-- DROP TABLE public.deanaccountdetails;

CREATE TABLE public.deanaccountdetails
(
    email character varying COLLATE pg_catalog."default" NOT NULL,
    password character varying COLLATE pg_catalog."default" NOT NULL,
    invitation_mail_body character varying COLLATE pg_catalog."default",
    invitation_mail_subject character varying COLLATE pg_catalog."default",
    send_notification_mail_body character varying COLLATE pg_catalog."default",
    send_notification_mail_subject character varying COLLATE pg_catalog."default",
    CONSTRAINT deanaccountdetailspk PRIMARY KEY (email)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.deanaccountdetails
    OWNER to postgres;
```

7. **tempstudent table**

```
-- Table: public.tempstudent

-- DROP TABLE public.tempstudent;

CREATE TABLE public.tempstudent
(
    email character varying COLLATE pg_catalog."default" NOT NULL,
    password character varying COLLATE pg_catalog."default" NOT NULL,
    supervisor_id character varying COLLATE pg_catalog."default" NOT NULL,
    research_area character varying COLLATE pg_catalog."default" NOT NULL,
    name character varying COLLATE pg_catalog."default" NOT NULL,
    token character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "tempstudentPk" PRIMARY KEY (email)
)
WITH (
    OIDS = FALSE
```

```
)
TABLESPACE pg_default;

ALTER TABLE public.tempstudent
    OWNER to postgres;
```

8. **thesisreviewer:**

```
-- Table: public.thesisreviewer

-- DROP TABLE public.thesisreviewer;

CREATE TABLE public.thesisreviewer
(
    thesis_id integer NOT NULL,
    reviewer_id character varying COLLATE pg_catalog."default" NOT NULL,
    supervisor_id character varying COLLATE pg_catalog."default" NOT NULL,
    token character varying COLLATE pg_catalog."default",
    sentdate date,
    status character varying COLLATE pg_catalog."default" NOT NULL,
    acceptteddate date,
    CONSTRAINT "thesisreviewerPK" PRIMARY KEY (thesis_id, reviewer_id),
    CONSTRAINT "reviewerFK" FOREIGN KEY (reviewer_id, supervisor_id)
        REFERENCES public.reviewer (email_id, supervisor_id) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT "thesisIdFK" FOREIGN KEY (thesis_id)
        REFERENCES public.thesis (thesis_id) MATCH SIMPLE
        ON UPDATE NO ACTION
```

```
        ON DELETE CASCADE
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.thesisreviewer
    OWNER to postgres;
```