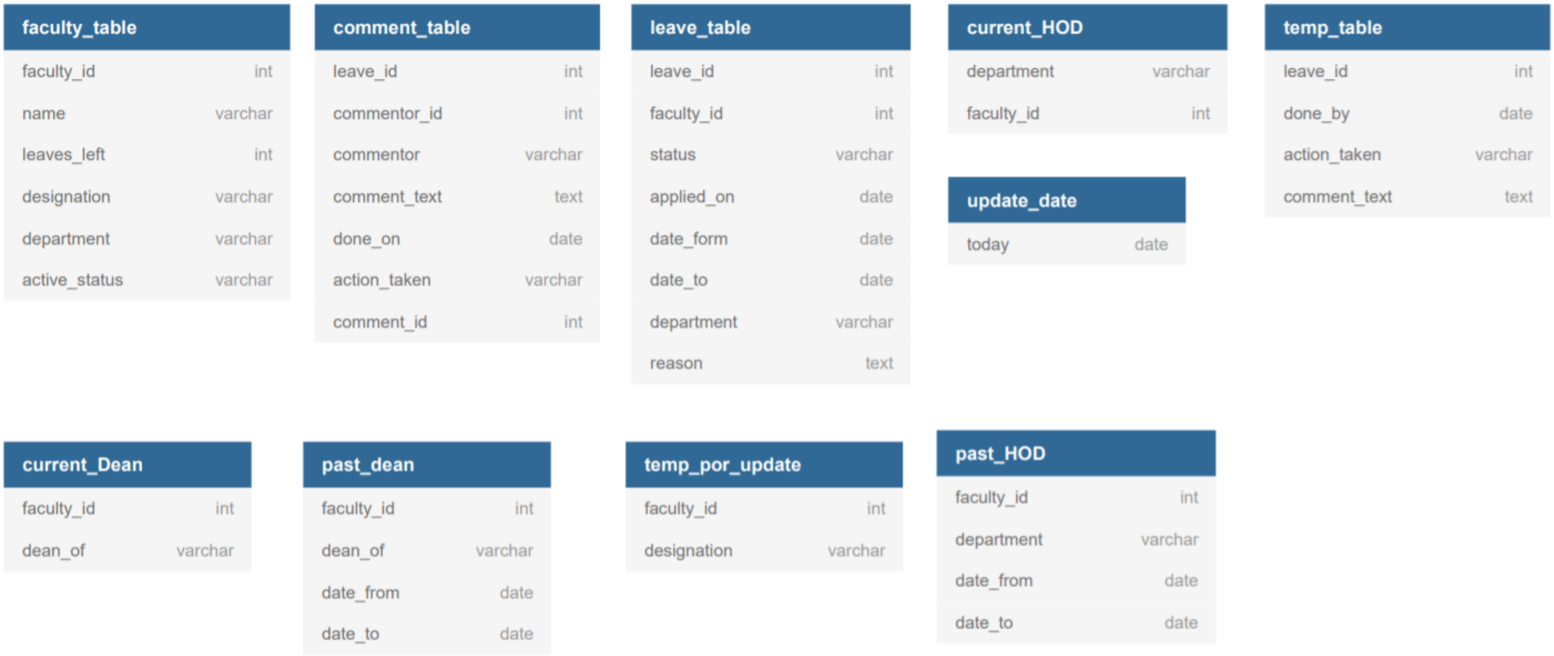
Submitted By : Ashutosh Garg(2018meb1213) and Ayush Agarwal(2018meb1215)



faculty_table	
faculty_id	int
name	varchar
leaves_left	int
designation	varchar
department	varchar
active_status	varchar

comment_table	
leave_id	int
commentor_id	int
commentor	varchar
comment_text	text
done_on	date
action_taken	varchar
comment_id	int

leave_table	
leave_id	int
faculty_id	int
status	varchar
applied_on	date
date_form	date
date_to	date
department	varchar
reason	text

current_HOD	
department	varchar
faculty_id	int

update_date	
today	date

temp_table	
leave_id	int
done_by	date
action_taken	varchar
comment_text	text

current_Dean	
faculty_id	int
dean_of	varchar

past_dean	
faculty_id	int
dean_of	varchar
date_from	date
date_to	date

temp_por_update	
faculty_id	int
designation	varchar

past_HOD	
faculty_id	int
department	varchar
date_from	date
date_to	date

Flow of the processes:

1. **Registering a faculty**
   Faculty registers > Details entered in *faculty* table through trigger to generate *faculty_id* and other values
2. **Leave application**
   Values like reason, start date and end date are entered in *leave_table* > trigger *new_leave_request* gets fired to generate *leave_id* and set status to pending with HOD or director according to designation > if less leaves are available than applied leaves, end date is adjusted through the trigger.
3. **New Comment**
   The comment values of commenter's id, leave id, action taken and comment text are input into a temporary table *temp_table* > trigger *new_update* is fired > status in *leave_table* is appropriately updates according to action taken, comment is inserted into *comment_table,* if dean has passed, then if retrospective or not is checked and then appropriately status is updated to pending with director or accepted and leaves are deducted.
4. **POR Update**
   The designation and faculty ID are input in *temp_porupdate* table > trigger *por_update* is fired. According to the designation, the dean/HOD is updated in *current_HOD / current_dean* and the previous POR holder is entered in *past_hods/past_dean* to maintain the paper trail, if the new POR holder has any pending leaves, they are automatically forwarded to director.
5. **Tasks on date updation**
   A table with a single cell was kept to maintain the last updated date called *update_date.* If a change in date took place, the trigger *change_indate* works and checks if any pending leave has reached its start date and not yet approved (if leave is retrospective then it is ignored). If any leave is rejected, then appropriate comment is also inserted in the *comment_table* from the system. Also, if new year is reached, then all the number of left leaves is reset to default value.

Register

```
create or replace function register_faculty()
returns trigger
language plpgsql
as $$
        begin
        new.faculty_id := max(faculty_id) from faculty;
        new.faculty_id := new.faculty_id +1;
        if(new.designation = 'HOD') then
            new.leaves_left := 25;
        elsif(new.designation = 'Dean Faculty Affairs') then
            new.leaves_left := 25;
        else
            new.leaves_left :=  25;
        end if;
        new.active := 'Active';
        return new;
end $$

create trigger register_faculty_trigger
        before insert on faculty
        for each row
            execute procedure register_faculty();
```

TESTING Registering a faculty

```
insert into faculty values (-1, 'Dr. Tyagi', -1, 'Faculty', 'ME')

select * from faculty
```

New Leave Request

```
create trigger new_leave_request
before insert on leaves
for each row execute procedure newleave();

create or replace function newleave()
returns trigger
language plpgsql
as
$$
declare
duration int := new.date_to - new.date_from;
comid int:= max(comment_id) from comment_table;

--leaveid int := max(leave_id) from leaves;
begin
            comid := comid+1;

if(comid is NULL) then
        comid := 1;
        end if;
        new.leave_id := max(leave_id) from leave_table;
        new.leave_id := new.leave_id +1;
        new.department := department from faculty where faculty.faculty_id = new.faculty_id;
        if duration = 0 then
            raise exception 'Invalid duration';
        elsif (select leaves_left from faculty where faculty.faculty_id = new.faculty_id)=0 then
            raise exception 'No leaves available';
        elsif duration > (select leaves_left from faculty where faculty.faculty_id = new.faculty_id) then
            if (select designation from faculty where faculty.faculty_id = new.faculty_id) = 'Faculty' then
                  new.date_to := new.date_from + (select leaves_left from faculty where faculty.faculty_id = new.faculty_id);
                  new.status := 'Pending with HOD';
                  raise notice 'Less leaves available, maximum leaves will be deducted';
            else
                  new.date_to := new.date_from + (select leaves_left from faculty where faculty.faculty_id = new.faculty_id);
                  new.status := 'Pending with Director';
                  raise notice 'Less leaves available, maximum leaves will be deducted';
            end if;
        else
            if (select designation from faculty where faculty.faculty_id = new.faculty_id) = 'Faculty' then
                  new.status := 'Pending with HOD';
                  --raise notice "Less leaves available, maximum leaves will be deducted";
            else
                  new.status := 'Pending with Director';
                  --raise notice "Less leaves available, maximum leaves will be deducted";
            end if;
```

```
leave_id int,
commentor_id int,
commentor varchar(50),
comment_text varchar(500),
done_on date,
action_taken varchar(50)
```

```
                end if;

                --EXECUTE format('CREATE TABLE %s (commentor_id int,commentor varchar(200), comment varchar(500), done_on date, action varchar(500))' ,CONCAT('comment_table', new.leave_id));
                EXECUTE format('insert into %s values(%s,%L, %L, %L, %L, %L, %L)' ,CONCAT('comment_table'),new.leave_id, new.faculty_id,'Applicant',new.reason,current_date, 'Applied', comid);

                return new;
end
$$


insert into leaves( faculty_id ,applied_on ,date_from ,date_to, reason )
values( 6, Current_date, '2021-05-06', '2021-05-09', 'Ma beemar hai')



Comment new
insert into temp_table values(leave id, faculty id who comments, action taken, comment_text)
alter table comment_table
add column comment_id int;

create or replace function new_comment()
returns trigger
language plpgsql
as
$$
declare

applicator varchar(20) := designation from faculty where faculty.faculty_id = (select leave_table.faculty_id from leave_table where leave_table.leave_id = new.leave_id);
applicant_id int := leave_table.faculty_id from leave_table where leave_table.leave_id = new.leave_id;
commentor varchar(20) := designation from faculty where faculty.faculty_id = new.done_by;
cname varchar(50) := faculty.faculty_name from faculty where faculty.faculty_id = new.done_by;
duration int := (leave_table.date_to - leave_table.date_from) from leave_table where leave_table.leave_id = new.leave_id;
comid int:= max(comment_id) from comment_table;
app date:=applied_on from leave_table where leave_id=new.leave_id;
fromt date:= date_from from leave_table where leave_id=new.leave_id;
--leaveid int := max(leave_id) from leave_table;
begin
                        comid := comid+1;
                        if(comid is NULL) then
                                    comid := 1;
                                    end if;
if commentor = 'Faculty' then
                update leave_table set status = 'Pending with HOD' where leave_id = new.leave_id;
                EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id, new.done_by , cname ,new.comment_text,current_date, 'Resent to HOD', comid);

elsif commentor = 'HOD' then
                if applicator = 'HOD' then
                        update leave_table set status = 'Pending with Director' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id, new.done_by , cname ,new.comment_text,current_date, 'Resent to Director', comid);
                elsif new.action_taken = 'Accepted' then
                        update leave_table set status = 'Pending with Dean FA' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id , new.done_by , cname ,new.comment_text,current_date, 'Sent to Dean FA', comid);
                elsif new.action_taken = 'Resent for corrections' then
                        update leave_table set status = 'Clarification pending' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id, new.done_by , cname ,new.comment_text,current_date, 'Resent for clarification', comid);
                elsif new.action_taken = 'Rejected' then
                        update leave_table set status = 'Rejected' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id , new.done_by , cname ,new.comment_text,current_date, 'Rejected', comid);
                end if;


elsif commentor = 'DOFA' then
                if applicator = 'DOFA' then
                        update leave_table set status = 'Pending with Director' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id, new.done_by , cname ,new.comment_text,current_date, 'Resent to Director', comid);
                elsif new.action_taken = 'Accepted' then
                        if app >= fromt then
                                    update leave_table set status = 'Pending with Director' where leave_id = new.leave_id;
                                    EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id , new.done_by , cname ,new.comment_text,current_date, 'Sent to Director', comid);
                        else
                                    update leave_table set status = 'Accepted' where leave_id = new.leave_id;
                                    update faculty set leaves_left = leaves_left- duration where faculty_id = applicant_id;
                                    EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id, new.done_by , cname ,new.comment_text,current_date, 'Accepted', comid);

                        end if;

                elsif new.action_taken = 'Resent for corrections' then
                        update leave_table set status = 'Clarification pending' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id , new.done_by , cname ,new.comment_text,current_date, 'Resent for clarification', comid);
                elsif new.action_taken = 'Rejected' then
                        update leave_table set status = 'Rejected' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id, new.done_by , cname ,new.comment_text,current_date, 'Rejected', comid);
                end if;

elsif commentor = 'Director' then
                if new.action_taken = 'Accepted' then
                                    update leave_table set status = 'Accepted' where leave_id = new.leave_id;
                                    update faculty set leaves_left = leaves_left- duration where faculty_id = applicant_id;
                                    EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id , new.done_by , cname ,new.comment_text,current_date, 'Accepted', comid);

                elsif new.action_taken = 'Resent for corrections' then
                        update leave_table set status = 'Clarification pending' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id, new.done_by , cname ,new.comment_text,current_date, 'Resent for clarification', comid);
                elsif new.action_taken = 'Rejected' then
                        update leave_table set status = 'Rejected' where leave_id = new.leave_id;
                        EXECUTE format('insert into comment_table values(%L,%s,%L, %L, %L, %L, %L)', new.leave_id , new.done_by , cname ,new.comment_text,current_date, 'Rejected', comid);
                end if;

end if;
return new;
end
$$


create trigger new_update
before insert on temp_table
for each row execute procedure new_comment();

insert into leaves( faculty_id ,applied_on ,date_from ,date_to, reason )
values( 12, Current_date, '2021-04-06', '2021-04-09', 'Ma beemar hai')

select * from faculty

insert into temp_table values(11, 1, 'Accepted', 'Ja beta ja')

select * from comment_table10


insert into faculty values(14, 'Prof. Rajeev Ahuja', 20, 'Director', 'EE', 1234567)
```

Automatically reject

```
update leave_table set status='Rejected' where (leave_table.applied_on<leave_table.date_from) and (leave_table.status <>'Accepted' and leave_table.status <>'Rejected')
and (leave_table.date_from <= current_date)
```

Automatically reset leaves

Update POR

```
insert into temp_porupdate values(4,'HOD ME')                    isme faculty ID aur POR daalni hai
create or replace function update_por()
returns trigger
language plpgsql
as
$$
declare
                previous int;
                leaveid int;
comid int:= max(comment_id) from comment_table;
begin

                        comid := comid+1;

                if (new.designation = 'HOD ME') then
                        if (select department from faculty where faculty.faculty_id = new.faculty_id)= 'ME' then
```

```
                    previous := faculty_id from faculty where faculty.designation = 'HOD' and faculty.department = 'ME';
                    update faculty set designation = 'Faculty' where faculty.faculty_id = previous;
                    update faculty set designation = 'HOD' where faculty.faculty_id = new.faculty_id;
                    update current_hod set faculty_id = new.faculty_id where current_hod.department='ME';
                    update past_hods set end_date = current_date where past_hods.faculty_id=previous and past_hods.end_date is null;
                    insert into past_hods values('ME', new.faculty_id, current_date);
                            update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    leaveid := leave_id from leave_table where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    insert into comment_table values (leaveid, 0, 'SYSTEM', 'Automatically sent to director on change in POR', current_date, 'Sent to Director', comid);
            else
            raise exception 'Different department';
            end if;
        elsif (new.designation = 'HOD EE') then
            if (select department from faculty where faculty.faculty_id = new.faculty_id)= 'EE' then
                    previous := faculty_id from faculty where faculty.designation = 'HOD' and faculty.department = 'EE';
                    update faculty set designation = 'Faculty' where faculty.faculty_id = previous;
                    update faculty set designation = 'HOD' where faculty.faculty_id = new.faculty_id;
                    update current_hod set faculty_id = new.faculty_id where current_hod.department='EE';
                    update past_hods set end_date = current_date where past_hods.faculty_id=previous and past_hods.end_date is null;
                    insert into past_hods values('EE', new.faculty_id, current_date);
                    update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
        update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    leaveid := leave_id from leave_table where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    insert into comment_table values (leaveid, 0, 'SYSTEM', 'Automatically sent to director on change in POR', current_date, 'Sent to Director', comid);

            else
            raise exception 'Different department';
            end if;
        --end if;
        elsif (new.designation = 'HOD CSE') then
            if (select department from faculty where faculty.faculty_id = new.faculty_id)= 'CSE' then
                    previous := faculty_id from faculty where faculty.designation = 'HOD' and faculty.department = 'CSE';
                    update faculty set designation = 'Faculty' where faculty.faculty_id = previous;
                    update faculty set designation = 'HOD' where faculty.faculty_id = new.faculty_id;
                    update current_hod set faculty_id = new.faculty_id where current_hod.department='CSE';
                    update past_hods set end_date = current_date where (past_hods.faculty_id=previous) and (past_hods.end_date is  NULL);
                    insert into past_hods values('CSE', new.faculty_id, current_date);
                    update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
        update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    leaveid := leave_id from leave_table where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    insert into comment_table values (leaveid, 0, 'SYSTEM', 'Automatically sent to director on change in POR', current_date, 'Sent to Director', comid);

            else
            raise exception 'Different department';
            end if;
        --end if;
        elsif (new.designation = 'HOD CHE') then
            if (select department from faculty where faculty.faculty_id = new.faculty_id)= 'CHE' then
                    previous := faculty_id from faculty where faculty.designation = 'HOD' and faculty.department = 'CHE';
                    update faculty set designation = 'Faculty' where faculty.faculty_id = previous;
                    update faculty set designation = 'HOD' where faculty.faculty_id = new.faculty_id;
                    update current_hod set faculty_id = new.faculty_id where current_hod.department='CHE';
                    update past_hods set end_date = current_date where past_hods.faculty_id=previous and past_hods.end_date  is null;
                    insert into past_hods values('CHE', new.faculty_id, current_date);
                    update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
        update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    leaveid := leave_id from leave_table where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    insert into comment_table values (leaveid, 0, 'SYSTEM', 'Automatically sent to director on change in POR', current_date, 'Sent to Director', comid);

            else
            raise exception 'Different department';
            end if;
        elsif (new.designation = 'DOFA') then

                    previous := faculty_id from faculty where faculty.designation = 'DOFA' ;
                    update faculty set designation = 'Faculty' where faculty.faculty_id = previous;
                    update faculty set designation = 'DOFA' where faculty.faculty_id = new.faculty_id;
                    update current_dean set faculty_id = new.faculty_id where current_dean.dean_of='DOFA';
                    update past_dean set end_date = current_date where past_dean.faculty_id=previous and past_dean.end_date is null;
                    insert into past_dean values('DOFA', new.faculty_id, current_date);
                    update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
            update leave_table set status ='Pending with Director' where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    leaveid := leave_id from leave_table where faculty_id=new.faculty_id and status <> 'Accepted' and status <> 'Rejected';
                    insert into comment_table values (leaveid, 0, 'SYSTEM', 'Automatically sent to director on change in POR', current_date, 'Sent to Director', comid);

            end if;

return new;
```

```
end
$$
```

Update date

```
create or replace function date_changed()
returns trigger
language plpgsql
as
$$
declare
            cur cursor for select * from leave_table;
            details leave_table%rowtype;
begin
if old.today <> new.today then
            open cur;
            loop
                  fetch cur into details;
                  exit when not found;
                  if((details.applied_on<details.date_from) and (details.status <>'Accepted' and details.status <>'Rejected') and (details.date_from <= new.today)) then
                            update leave_table set status='Rejected' where leave_table.leave_id = details.leave_id;
                            insert into comment_table(leave_id,commentor_id, commentor, comment_text, done_on, action_taken) values(details.leave_id, 0, 'SYSTEM', 'Auto Rejected', new.today, 'Rejected');
                  end if;
            end loop;

            if (extract(month from new.today) = 1 and extract(day from new.today) = 1) then
                  update faculty set leaves_left=25;
            end if;
end if;
return new;
end
$$

create trigger change_indate
before update on update_date
for each row execute procedure  date_changed();

update update_date set today='2021-01-03'
```