```
--create employees table
 CREATE TABLE Employees (
Emp_ID INT PRIMARY KEY,
Name VARCHAR(100),
Salary DECIMAL(10, 2)
);

--create audit_log table
CREATE TABLE AuditLog (
LogID SERIAL PRIMARY KEY,
Action VARCHAR(50),
ActionDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
Details VARCHAR(255)
 );


--query1
--function
create or replace function add_emp_log()
returns trigger language plpgsql
as $$
begin
insert into auditlog(action,ActionDate,Details)values
('insert',now(),'added new employee name-' || new.name);
return new;
end;
$$;

--trigger
create trigger after_emp_insert
after insert on employees
for each row
execute function  add_emp_log();

--insert
INSERT INTO Employees (Emp_ID, Name, Salary) VALUES
 (1, 'Keerthi', 55000.00);
 emp_id |  name   |  salary
--------+---------+----------
      1 | Keerthi | 55000.00

--after insert
 logid | action |         actiondate         |             details
-------+--------+----------------------------+---------------------------------
     1 | insert | 2025-09-15 20:56:04.999956 | added new employee name-Keerthi
(1 row)
```

```
--after insert
 logid | action |         actiondate         |             details
-------+--------+----------------------------+---------------------------------
     1 | insert | 2025-09-15 20:56:04.999956 | added new employee name-Keerthi
     2 | insert | 2025-09-15 20:57:01.469865 | added new employee name-Bala

--insert
INSERT INTO Employees (Emp_ID, Name, Salary) VALUES
 (3, 'Kutti', 48000.00),
 (4, 'Nithish', 75000.00),
 (5, 'Rohit', 90000.00);
 emp_id |  name   |  salary
--------+---------+----------
      1 | Keerthi | 55000.00
      2 | Bala    | 65000.00
      3 | Kutti   | 48000.00
      4 | Nithish | 75000.00
      5 | Rohit   | 90000.00
(5 rows)

--after insert
 logid | action |         actiondate         |             details
-------+--------+----------------------------+---------------------------------
     1 | insert | 2025-09-15 20:56:04.999956 | added new employee name-Keerthi
     2 | insert | 2025-09-15 20:57:01.469865 | added new employee name-Bala
     3 | insert | 2025-09-15 20:59:11.20222  | added new employee name-Kutti
     4 | insert | 2025-09-15 20:59:11.20222  | added new employee name-Nithish
     5 | insert | 2025-09-15 20:59:11.20222  | added new employee name-Rohit                    n
(5 rows)
```

```
--query2
--function
create or replace function notify_emp()
returns trigger language plpgsql
as $$
begin
insert into auditlog(action,ActionDate,Details)values
('insert',now(),'added new employee name-' || new.name);

raise notice 'New employee added: %',new.name;
return new;
end;
$$;

--trigger
create trigger notify_emp_insert
after insert on employees
for each row
execute function  notify_emp();

--insert
insert into employees values
(6,'Surya',35000.00);

--after insert
NOTICE:  New employee added: Surya
INSERT 0 1
 logid | action |         actiondate         |             details
-------+--------+----------------------------+---------------------------------
     1 | insert | 2025-09-15 20:56:04.999956 | added new employee name-Keerthi
     2 | insert | 2025-09-15 20:57:01.469865 | added new employee name-Bala
     3 | insert | 2025-09-15 20:59:11.20222  | added new employee name-Kutti
     4 | insert | 2025-09-15 20:59:11.20222  | added new employee name-Nithish
     5 | insert | 2025-09-15 20:59:11.20222  | added new employee name-Rohit
     6 | insert | 2025-09-15 21:02:04.064327 | added new employee name-Surya
     7 | insert | 2025-09-15 21:02:04.064327 | added new employee name-Surya
(7 rows)
```

```
--query3
--function
create or replace function sal_update()
returns trigger
language plpgsql
as $$
begin
if new.salary<old.salary
then
raise exception 'employee salary cant be decreased ';
end if;
return new;
end;
$$;

--trigger
create trigger low_update_salary
before update on employees
for each row
execute function sal_update();

--update
 update employees
 set salary=20000.00
 where emp_id=1;
--after update
ERROR:  employee salary cant be decreased

--create employee_archive table
CREATE TABLE EmployeesArchive (
 Emp_ID INT,
 Name VARCHAR(100),
 Salary DECIMAL(10, 2),
 DeletedDate DATE DEFAULT current_date
);
```

```
);

--query4
--function
create or replace function insert_delete_emp()
returns trigger
language plpgsql
as
$$
begin
insert into employeesarchive
values(old.emp_id,old.name,old.salary,now());
return old;
end;
$$;

--trigger
create trigger del_emp_add
before delete on employees
for each row
execute function insert_delete_emp();

--delete
 delete from employees
 where emp_id=6;

--before delete
 emp_id | name  |  salary  | deleteddate
--------+-------+----------+-------------
      6 | Surya | 35000.00 | 2025-09-15

--after delete
 emp_id |  name   |  salary
--------+---------+----------
      1 | Keerthi | 55000.00
      2 | Bala    | 65000.00
      3 | Kutti   | 48000.00
      4 | Nithish | 75000.00
      5 | Rohit   | 90000.00
(5 rows)
```

```
--query5
--function
create or replace function del_emp()
returns trigger
language plpgsql
as
$$
begin
insert into employeesarchive
values(old.emp_id,old.name,old.salary,now());
return new;
end;
$$;

--trigger
create trigger emp_del
after delete on employees
for each row
execute function del_emp();

--delete
 delete from employees
 where emp_id=6;

--after delete
--employees
 emp_id |  name   |  salary
--------+---------+----------
      1 | Keerthi | 55000.00
      2 | Bala    | 65000.00
      3 | Kutti   | 48000.00
      4 | Nithish | 75000.00
      5 | Rohit   | 90000.00
(5 rows)

--employeesarchieve
 emp_id | name  |  salary  | deleteddate
--------+-------+----------+-------------
      6 | Surya | 35000.00 | 2025-09-15
      6 | Surya | 35000.00 | 2025-09-15
      6 | Surya | 35000.00 | 2025-09-15
(1 row)
```

```
--query 6
--function1
 create or replace function ins_emp_count()
 returns trigger
 language plpgsql
 as
 $$
 begin
 update count_tab
 set count=count+1;
 return new;
 end;
 $$;

--function2
 create or replace function del_emp_count()
 returns trigger
 language plpgsql
 as $$
 begin
 update count_tab
 set count=count-1;
 return new;
 end;
 $$;

--trigger1
create trigger trg_ins_emp_count
after insert on employees
for each row
execute function ins_emp_count();

--trigger2
 create trigger trg_ins_del_count
 after delete on employees
 for each row
 execute function del_emp_count();

--insert1
insert into employees values(6,'Surya',35000.00);
NOTICE:  New employee added: Surya
INSERT 0 1
--insert2
insert into employees values(7,'Yogi',45000.00);
NOTICE:  New employee added: Surya
INSERT 0 1

 count
-------
     2
```

```
--query 7
--function

--function
create or replace function log_employee_changes()
returns trigger
language plpgsql
as $$
begin
if TG_OP= 'INSERT' then
insert into AuditLog (action, ActionDate, Details)
values('INSERT', now(), 'Added new employee name - ' || new.name);
return new;
elseif TG_OP = 'UPDATE' then
if(new.salary is distinct from old.salary)
or(new.name is distinct from old.name) then
insert into AuditLog (action, ActionDate, Details)
values('UPDATE', now(),
                    'Updated employee id=' || NEW.id ||
                    ', name=' || NEW.name ||
                    ', salary=' || NEW.salary);
end if;
return new;
end if;
return NULL:
end;
$$;

-- Insert trigger
drop trigger if exists trg_log_employee_insert ON Employees;
create trigger trg_log_employee_insert
after insert on Employees
for each row
execute function log_employee_changes();

-- Update trigger
drop trigger if exists trg_log_employee_update ON Employees;
create trigger trg_log_employee_update
after update on Employees
for each row
execute function log_employee_changes();

-- Insert
insert into Employees (emp_id, name, salary)
values (7,'Yugan',72000.00);

-- Update
update Employees
set salary = 75000
where emp_id = 7;
```

```
-- Update
update Employees
set salary = 75000
where emp_id = 7;

-- Check log
select * from AuditLog;
```

log_id	action	actiondate	details
1	insert	2025-09-16 21:42:04.626272	added new employee name-Yugan
2	INSERT	2025-09-16 21:42:04.626272	Added new employee name - Yugan
3	insert	2025-09-16 21:42:04.626272	added new employee name-Prathiban
4	INSERT	2025-09-16 21:42:04.626272	Added new employee name - Prathiban
5	insert	2025-09-16 21:42:04.626272	added new employee name-Vignesh
6	INSERT	2025-09-16 21:42:04.626272	Added new employee name - Vignesh
7	insert	2025-09-16 21:42:04.626272	added new employee name-Sundar
8	INSERT	2025-09-16 21:42:04.626272	Added new employee name - Sundar

```
--my domain
--create staffaudit table
CREATE TABLE StaffAudit (
    audit_id   SERIAL PRIMARY KEY,
    staff_id   INT NOT NULL,
    action     TEXT NOT NULL,
    action_date TIMESTAMP DEFAULT now()
);

--function
create or replace function ensure_one_hod()
returns trigger
language plpgsql
as $$
begin
if new.hod_id = new.staff_id then
update Staff
set hod_id = NULL
where dept_id = NEW.dept_id
and hod_id = staff_id
and staff_id <> new.staff_id;
insert into StaffAudit(staff_id, action, action_date)
values (NEW.staff_id, 'BECAME HOD', now());
end if;
return new;
end;
$$;

--trigger
create trigger trg_one_hod
before insert or update on Staff
for each row
execute function ensure_one_hod();

--insert
insert into Staff (staff_id, staff_name, start_date, dept_id, hod_id)
VALUES (1018, 'Kumaravel', '2025-01-01', 1, 1018);

--select
select staff_id, staff_name, dept_id, hod_id
from Staff
where dept_id = 1;
--op
 audit_id | staff_id |   action    |        action_date
----------+----------+-------------+----------------------------
        1 |     1018 | BECAME HOD | 2025-09-16 21:58:55.895686
(1 row)
```

```
--query 2

--create table

alter table Staff
add column if not exists years_of_service int;
create table if not exists StaffRetirementLog (
    log_id serial primarykey,
    staff_id int notnull,
    action text notnull,
    action_date timestamp default now()
);

--function
create or replace staff_retirement_update()
returns trigger
language plpgsql
as $$
begin
if new.retired_date is not null and (old.retired_date is distinct from new.retired_date) then
new.years_of_service := extract(year from age(new.retired_date, new.start_date))::int;
insert into StaffRetirementLog(staff_id, action, action_date)
values(new.staff_id, 'Retired', now());
end if;
return new;
end;
$$;

--trigger
drop trigger if not exists trg_staff_retirement ON Staff;

create trigger trg_staff_retirement
before update on Staff
for each row
execute function staff_retirement_update();

-- Update retired_date for a staff
UPDATE Staff
SET retired_date = '2025-12-31'
WHERE staff_id = 1006;

--select
select staff_id, start_date, retired_date, years_of_service from Staff
where staff_id = 1006;

select * from StaffRetirementLog;

 log_id | staff_id | action  |        action_date
--------+----------+---------+----------------------------
      1 |     1006 | Retired | 2025-09-16 22:30:20.917447
(1 row)
```