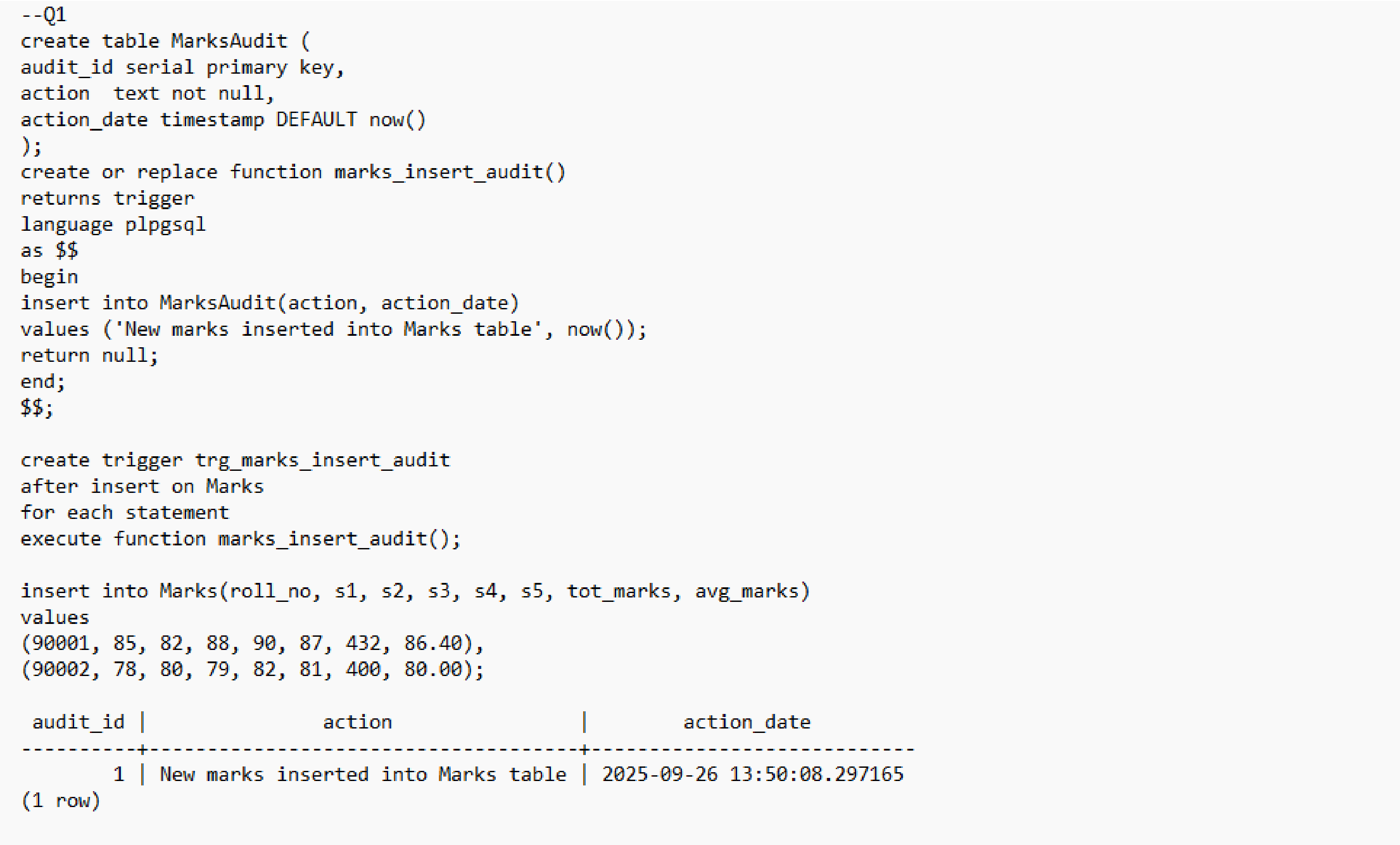
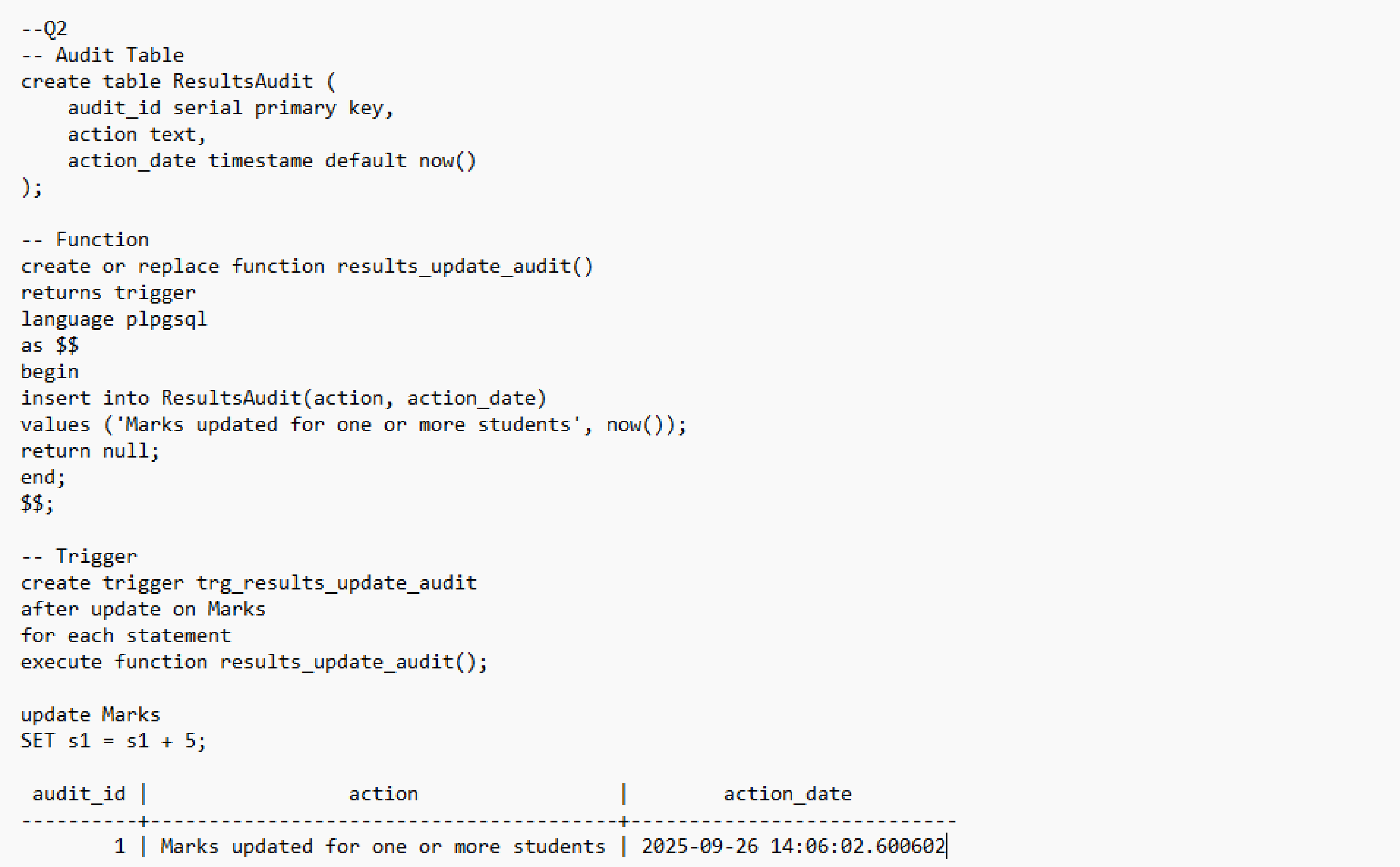


```
--Q1
create table MarksAudit (
audit_id serial primary key,
action  text not null,
action_date timestamp DEFAULT now()
);
create or replace function marks_insert_audit()
returns trigger
language plpgsql
as $$
begin
insert into MarksAudit(action, action_date)
values ('New marks inserted into Marks table', now());
return null;
end;
$$;

create trigger trg_marks_insert_audit
after insert on Marks
for each statement
execute function marks_insert_audit();

insert into Marks(roll_no, s1, s2, s3, s4, s5, tot_marks, avg_marks)
values
(90001, 85, 82, 88, 90, 87, 432, 86.40),
(90002, 78, 80, 79, 82, 81, 400, 80.00);

 audit_id |               action                |        action_date
----------+-------------------------------------+----------------------------
        1 | New marks inserted into Marks table | 2025-09-26 13:50:08.297165
(1 row)
```



```
--Q2
-- Audit Table
create table ResultsAudit (
    audit_id serial primary key,
    action text,
    action_date timestame default now()
);

-- Function
create or replace function results_update_audit()
returns trigger
language plpgsql
as $$
begin
insert into ResultsAudit(action, action_date)
values ('Marks updated for one or more students', now());
return null;
end;
$$;

-- Trigger
create trigger trg_results_update_audit
after update on Marks
for each statement
execute function results_update_audit();

update Marks
SET s1 = s1 + 5;

 audit_id |                 action                 |        action_date
----------+----------------------------------------+----------------------------
        1 | Marks updated for one or more students | 2025-09-26 14:06:02.600602
```