**ΕΡΩΤΗΜΑ 2.**

CREATE FUNCTION BetterPlaceThanPosition() RETURNS TRIGGER AS $$

BEGIN

IF NEW.raceposition < NEW.grid THEN

UPDATE f1\_driver

SET points=points + 5

WHERE code = NEW.drivercode;

END IF;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER BetterPlace

AFTER INSERT ON f1\_results

FOR EACH ROW

EXECUTE PROCEDURE BetterPlaceThanPosition();

**ΕΡΩΤΗΜΑ 3.**

CREATE FUNCTION PutPoint() RETURNS TRIGGER AS $$

BEGIN

IF NEW.raceposition = 1 THEN

UPDATE f1\_driver

SET points=points + 25

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 2 THEN

UPDATE f1\_driver

SET points=points + 18

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 3 THEN

UPDATE f1\_driver

SET points=points + 15

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 4 THEN

UPDATE f1\_driver

SET points=points + 12

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 5 THEN

UPDATE f1\_driver

SET points=points + 10

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 6 THEN

UPDATE f1\_driver

SET points=points + 8

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 7 THEN

UPDATE f1\_driver

SET points=points + 6

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 8 THEN

UPDATE f1\_driver

SET points=points + 4

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 9 THEN

UPDATE f1\_driver

SET points=points + 2

WHERE code = NEW.drivercode;

ELSIF NEW.raceposition = 10 THEN

UPDATE f1\_driver

SET points=points + 1

WHERE code = NEW.drivercode;

END IF;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER Points

AFTER INSERT ON f1\_results

FOR EACH ROW

EXECUTE PROCEDURE PutPoint();

**ΕΡΩΤΗΜΑ 4.**

CREATE TABLE f1\_results\_log\_file (

operation char(1) NOT NULL,

oper\_time timestamp NOT NULL,

drivercode varchar(5) NOT NULL,

circuitcode varchar(5) NOT NULL,

racedate date,

grid integer,

primary key(drivercode, racedate)

);

**ΕΡΩΤΗΜΑ 5.**

CREATE OR REPLACE FUNCTION createlog ()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO f1\_results\_log\_file SELECT 'D', now(), OLD.drivercode, OLD.circuitcode, OLD.racedate, OLD.grid;

RETURN OLD;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO f1\_results\_log\_file SELECT 'U', now(), NEW.drivercode, NEW.circuitcode, NEW.racedate, NEW.grid;

RETURN NEW;

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO f1\_results\_log\_file SELECT 'I', now(), NEW.drivercode, NEW.circuitcode, NEW.racedate, NEW.grid;

RETURN NEW;

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER keeplog

AFTER INSERT OR UPDATE OR DELETE ON f1\_results

FOR EACH ROW EXECUTE PROCEDURE createlog();

insert into f1\_results values ('D1', 'CI1', date('2015/3/6'), 3, 13);

insert into f1\_results values ('D1', 'CI1', date('2016/3/6'), 3, 13);

UPDATE f1\_results SET grid = 2 WHERE drivercode = 'D2';

DELETE from f1\_results Where drivercode = 'D3';