--1

CREATE OR REPLACE FUNCTION showQ1() RETURNS SETOF vSt AS $$

SELECT \* FROM vSt;

$$ LANGUAGE sql;

--2

CREATE OR REPLACE FUNCTION showQ2() RETURNS SETOF vTe AS $$

SELECT \* FROM vTe;

$$ LANGUAGE sql;

--3

CREATE OR REPLACE FUNCTION showQ3() RETURNS SETOF vCo AS $$

SELECT \* FROM vCo;

$$ LANGUAGE sql;

--4

CREATE OR REPLACE FUNCTION showQ4() RETURNS SETOF vCl AS $$

SELECT \* FROM vCl;

$$ LANGUAGE sql;

--5

CREATE OR REPLACE FUNCTION showQ5() RETURNS SETOF vSche AS $$

SELECT \* FROM vSche;

$$ LANGUAGE sql;

--6

CREATE OR REPLACE FUNCTION showQ6() RETURNS SETOF vQ6 AS $$

SELECT \* FROM vQ6;

$$ LANGUAGE sql;

--7

CREATE OR REPLACE FUNCTION showQ7() RETURNS SETOF vQ7 AS $$

SELECT \* FROM vQ7;

$$ LANGUAGE sql;

--8

CREATE OR REPLACE FUNCTION showQ8() RETURNS SETOF vQ8 AS $$

SELECT \* FROM vQ8;

$$ LANGUAGE sql;

--9

CREATE OR REPLACE FUNCTION showQ9() RETURNS SETOF vQ9 AS $$

SELECT \* FROM vQ9;

$$ LANGUAGE sql;

--10

CREATE OR REPLACE FUNCTION showQ10() RETURNS SETOF vQ10 AS $$

SELECT \* FROM vQ10;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION process\_st\_audit ()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO audit SELECT 'D', now(), user, OLD.sid, 'Pupil';

RETURN OLD;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO audit SELECT 'U', now(), user, NEW.sid, 'Pupil';

RETURN NEW;

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO audit SELECT 'I', now(), user, NEW.sid, 'Pupil';

RETURN NEW;

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER audit

AFTER INSERT OR UPDATE OR DELETE ON Students

FOR EACH ROW EXECUTE PROCEDURE process\_st\_audit();

CREATE OR REPLACE FUNCTION process\_te\_audit ()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO audit SELECT 'D', now(), user, OLD.tid, 'Professor';

RETURN OLD;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO audit SELECT 'U', now(), user, NEW.tid, 'Professor';

RETURN NEW;

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO audit SELECT 'I', now(), user, NEW.tid, 'Professor';

RETURN NEW;

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER audit

AFTER INSERT OR UPDATE OR DELETE ON Teachers

FOR EACH ROW EXECUTE PROCEDURE process\_te\_audit();

CREATE OR REPLACE FUNCTION process\_co\_audit ()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO audit SELECT 'D', now(), user, OLD.cid, 'Lesson';

RETURN OLD;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO audit SELECT 'U', now(), user, NEW.cid, 'Lesson';

RETURN NEW;

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO audit SELECT 'I', now(), user, NEW.cid, 'Lesson';

RETURN NEW;

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER audit

AFTER INSERT OR UPDATE OR DELETE ON Courses

FOR EACH ROW EXECUTE PROCEDURE process\_co\_audit();

CREATE OR REPLACE FUNCTION process\_cl\_audit ()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO audit SELECT 'D', now(), user, OLD.classId, 'classRoom';

RETURN OLD;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO audit SELECT 'U', now(), user, NEW.classId, 'classRoom';

RETURN NEW;

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO audit SELECT 'I', now(), user, NEW.classId, 'classRoom';

RETURN NEW;

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER audit

AFTER INSERT OR UPDATE OR DELETE ON Classes

FOR EACH ROW EXECUTE PROCEDURE process\_cl\_audit();

CREATE OR REPLACE FUNCTION process\_gr\_audit ()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO audit SELECT 'D', now(), user, OLD.sid, 'Vathmos';

RETURN OLD;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO audit SELECT 'U', now(), user, NEW.sid, 'Vathmos';

RETURN NEW;

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO audit SELECT 'I', now(), user, NEW.sid, 'Vathmos';

RETURN NEW;

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER audit

AFTER INSERT OR UPDATE OR DELETE ON Grades

FOR EACH ROW EXECUTE PROCEDURE process\_gr\_audit();

CREATE OR REPLACE FUNCTION process\_sche\_audit ()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'DELETE') THEN

INSERT INTO audit SELECT 'D', now(), user, OLD.classId, 'Calendar(id of classRoom)';

RETURN OLD;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO audit SELECT 'U', now(), user, NEW.classId, 'Calendar(id of classRoom)';

RETURN NEW;

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO audit SELECT 'I', now(), user, NEW.classId, 'Calendar(id of classRoom)';

RETURN NEW;

END IF;

RETURN NULL;

END;

$$

LANGUAGE plpgsql;

CREATE TRIGGER audit

AFTER INSERT OR UPDATE OR DELETE ON Schedule

FOR EACH ROW EXECUTE PROCEDURE process\_sche\_audit();

CREATE OR REPLACE FUNCTION logFile() RETURNS SETOF iee2019067.audit AS $$

SELECT \* FROM iee2019067.audit ORDER BY stamp;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION search1(p1 in varchar) RETURNS SETOF Students AS

$$

select \* from Students

where ssurname = $1;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION search2(p1 in varchar) RETURNS SETOF Teachers AS $$

SELECT \* FROM Teachers

where tsurname = $1;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION search3(p1 in varchar) RETURNS SETOF Schedule AS $$

SELECT \* FROM Schedule

where date = $1;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION search4(p1 in varchar) RETURNS SETOF Classes AS $$

SELECT \* FROM Classes

where className = $1;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION search5(p1 in varchar) RETURNS SETOF Courses AS $$

SELECT \* FROM Courses

where cname = $1;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION search6(p1 in integer) RETURNS SETOF Grades AS $$

SELECT \* FROM Grades

where sid = $1;

$$ LANGUAGE sql;