--creacio de taula

DROP TABLE IF EXISTS treballadors;

CREATE TABLE treballadors {

id INT GENERATED ALWAYS AS IDENTITY,

nom VARCHAR(40) NOT NULL,

cognoms VARCHAR(40) NOT NULL,

PRIMARY KEY(id)

};

CREATE OR REPLACE FUNCTION trigger1() --no demana parametres d’entrada i sempre funcio

returns trigger --retorna sempre un trigger

language plpgsql

AS

$$

BEGIN

IF new.cognoms <> OLD.cognoms THEN

INSERT INTO treballador\_revisions(treballador\_id, cognoms,modificat)

VALUES(OLD.id, OLD.cognoms, now());

END IF;

RETURN NEW;

END;

$$

DROP TABLE IF EXISTS treballadors;

CREATE TABLE treballadors (

id INT GENERATED ALWAYS AS IDENTITY,

nom varchar(40) NOT NULL,

cognoms varchar(80) NOT NULL,

PRIMARY KEY (id));

CREATE TABLE treballador\_revisions(

id INT GENERATED ALWAYS AS IDENTITY,

treballador\_id INT NOT NULL,

cognoms VARCHAR(80) NOT NULL,

modificat TIMESTAMP(6) NOT NULL

);

Esquelet generic:

CREATE TRIGGER nom\_trigger

BEFORE | AFTER {event}

ON nom\_taula

FOR EACH {ROW | STATEMENT}

EXECUTE PROCEDURE function\_trigger

1) Trigger : trigger\_crear\_email\_before\_insert

- S'executa sobre la taula alumnes.

- S'executa abans d'una operació d'inserció.

- Si el nou valor del correu que es vol inserir és NULL, llavors se li crearà automàticament una adreça de correu electrònic i s'inserirà en la taula.

NOTA: L’adreça de correu tindrà el següent format: el primer carácter del nom, els tres primers caràcters del cognom1, els tres primers caràcters del cognom2, el carácter “@” i a continuació un domini fix que podeu escollir lliurement, com per exemple: iesbenigaslo.es

- Si el nou valor del correu NO és NULL, es guarda en la taula el valor de l’email.

id\_alumne: id de l'alumne (sencer)

data\_hora: marca de temps amb l'instant del canvi (data i hora)

nom: nom de l’alumne eliminat (cadena de caràcters)

cognom1: 1r cognom de l’alumne eliminat (cadena de caràcters)

cognom2: 2n cognom de l’alumne eliminat (cadena de caràcters)

email: correu de l’alumne eliminat (cadena de caràcters)

1:

DROP TABLE IF EXISTS alumnes;

CREATE TABLE alumnes (

id\_alumne SERIAL,

nom varchar(15),

cognom1 varchar(15),

cognom2 varchar(15),

email varchar(50));

CREATE OR REPLACE FUNCTION email()

returns trigger

LANGUAGE PLPGSQL

AS

$$

DECLARE

domini constant varchar(120) := '@iesbenigaslo.es';

BEGIN

IF NEW.email IS NULL THEN

NEW.email = LOWER(substring(NEW.nom,1,1), substring(NEW.cognom1,1,3),

substring(NEW.cognom2,1,3)) --AGAFAREM LA PRIMERA LLETRA DEL NOM I LES 3 PRIMERES DELS COGNOMS AMB EL SUBSTRING;

NEW.email := LOWER(email);

END IF;

RETURN NEW;

END;

$$

CREATE TRIGGER defaultEmail

BEFORE INSERT OR UPDATE

ON alumnes

FOR EACH ROW

EXECUTE PROCEDURE email();

2) Trigger: trigger\_guardar\_email\_after\_update

- S'executa sobre la taula alumnes.

- S'executa després d'una operació d'actualització.

- Cada vegada que un alumne modifique la seua adreça de correu, s’haurà d’inserir un nou registre en una taula anomenada log\_canvis\_email.

NOTA: La taula log\_canvis\_email conté els següents camps:

id: clau primària (sencer autonumèric)

id\_alumne: id de l'alumne (sencer)

data\_hora: marca de temps amb l'instant del canvi (data i hora)

old\_email: valor anterior de l'email (cadena de caràcters)

new\_email: nou valor amb el qual s'ha actualitzat (cadena de caràcters)

CREATE TABLE canvis\_email(

ide SERIAL PRIMARY KEY,

id\_alumne integer,

data\_hora timestamp,

old\_email varchar(50),

new\_email varchar(50)

)

CREATE OR REPLACE FUNCTION trigger2()

RETURNS TRIGGER;

LANGUAGE PLPGSQL

BEGIN

$$

IF NEW.email != OLD.email

THEN

INSERT INTO canvis\_email (id\_alumne, nom, data\_hora, old\_email, new\_email)

VALUES(OLD.id, OLD.nom, now(), OLD.email, NEW.email);

END IF;

RETURN NEW,

END;

$$;

CREATE TRIGGER trigger2

AFTER UPDATE

ON alumnes

FOR EACH ROW

EXECUTE PROCEDURE trigger2;

3) Trigger: trigger\_guardar\_alumnes\_eliminats

- S'executa sobre la taula alumnes.

- S'executa després d'una operació d'esborrat.

- Cada vegada que s’elimine un alumne de la taula alumnes, s’haurà d’inserir un nou registre en una taula anomenada log\_alumnes\_eliminats.

NOTA: La taula log\_alumnes\_eliminats conté els següents camps:

id: clau primària (sencer autonumèric)

CREATE OR REPLACE FUNCTION trigger3()

RETURNS TRIGGER

LANGUAGE PLPGSQL

AS

BEGIN

$$

INSERT INTO canvis\_email(DEFAULT, old.nom,old.cognom1, old.cognom2, old.email);

RETURN NEW;

END;

$$

CREATE TRIGGER trigger3\_

AFTER DELETE

ON alumnes

FOR EACH ROW

EXECUTE PROCEDURE trigger3;