



Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Monterrey

TC2005B.573

Queries Base De Datos Entrega Final

Luis Alberto Rodríguez Mendoza	A00828335
Rodolfo Sandoval Schipper	A01720253
Tadeo Andrés Willis Lozano	A00829341
Andrea Garza	A00832444

19/10/2022

Queries (base De datos).-

--TABLAS

```
CREATE TABLE users (  
    nombre varchar(49),  
    apellidop varchar(50),  
    apellidom varchar(50),  
    edad int,  
    email varchar(50),  
    passw varchar(30),  
    vali bool,  
    u_id serial PRIMARY KEY  
)
```

```
CREATE TABLE cursos (  
    nombre_c varchar(30),  
    descripcion varchar(255),  
    email varchar(50),  
    curso_id serial  
)
```

```
CREATE TABLE bitacora (  
    description VARCHAR(90) NOT NULL,  
    created_at TIMESTAMP DEFAULT NOW()  
);
```

-- FIN DE LAS TABLAS

-- STORED PROCEDURES

```
CREATE OR REPLACE PROCEDURE user_edit (nombre varchar(49), apellidop
varchar(50), apellidom varchar(50), edad int, email varchar(50), passw varchar(30), u_id int)
AS $$
    UPDATE users SET nombre = $1, apellidop = $2, apellidom = $3, edad = $4, email =
    $5, passw = $6
    WHERE u_id = $7
$$ LANGUAGE SQL
```

```
CREATE OR REPLACE PROCEDURE user_del (u_id int)
AS $$
    DELETE FROM users WHERE u_id = $1
$$ LANGUAGE SQL
```

```
CREATE OR REPLACE PROCEDURE user_insert (nombre varchar(49), apellidop
varchar(50), apellidom varchar(50), edad int, email varchar(50), passw varchar(30), vali bool,
u_id int)
AS $$
    INSERT INTO users VALUES(nombre, apellidop,apellidom,edad,email, passw,
vali,u_id)
$$ LANGUAGE SQL
```

```
CREATE OR REPLACE PROCEDURE insert_course( nombre_c varchar(50), descripcion
varchar(140), email varchar(50))
AS $$
    INSERT INTO cursos VALUES(nombre_c, descripcion, email)
$$ LANGUAGE SQL
```

```
CREATE OR REPLACE PROCEDURE guardar_usuario (nombre varchar(49), apellidop
varchar(50), apellidom varchar(50), edad int, email varchar(50), passw varchar(30), vali bool)
AS $$
    INSERT INTO users VALUES(nombre, apellidop,apellidom,edad,email, passw, vali)
```

```
$$ LANGUAGE SQL
```

```
-- FIN DE STORED PROCEDURES
```

```
-- INICIO DE FUNCTIONS
```

```
CREATE OR REPLACE FUNCTION fn_validar(lg_email varchar(50),lg_passw varchar(30))  
    RETURNS TABLE(bandera int)
```

```
AS $$
```

```
    SELECT COUNT (email) FROM users WHERE email = lg_email AND passw  
= lg_passw
```

```
$$ LANGUAGE SQL
```

```
CREATE OR REPLACE FUNCTION fn_admin(lg_email varchar(50), lg_passw  
varchar(30),lg_vali bool)
```

```
    RETURNS TABLE(admin_vali int)
```

```
AS $$
```

```
    SELECT COUNT(email) FROM users WHERE email = lg_email AND passw  
= lg_passw AND vali = lg_vali
```

```
$$ LANGUAGE SQL
```

```
CREATE OR REPLACE FUNCTION fn_consultar ()
```

```
    RETURNS TABLE(nombre varchar(49), apellidop varchar(50), apellidom varchar(50),  
edad int, email varchar(50), passw varchar(30), vali bool, u_id int)
```

```
AS $$
```

```
    SELECT a.nombre, a.apellidop, a.apellidom, a.edad, a.email, a.passw, a.vali, a.u_id  
FROM users AS a
```

```
$$ LANGUAGE SQL
```

```
CREATE OR REPLACE FUNCTION fn_obtener(u_id int)
```

```
    RETURNS TABLE (nombre varchar(49), apellidop varchar(50), apellidom varchar(50),  
edad int, email varchar(50), passw varchar(30), vali bool, u_id int)
```

```
AS $$  
SELECT a.nombre, a.apellidop, a.apellidom, a.edad, a.email, a.passw, a.vali, a.u_id  
FROM users AS a WHERE u_id = $1  
$$ LANGUAGE SQL
```

```
-- FIN DE FUNCIONES
```

```
-- INICIO TRIGGER FUNCTIONS
```

```
CREATE OR REPLACE FUNCTION fn_after_insert_user()  
RETURNS TRIGGER AS $BODY$  
BEGIN  
    INSERT INTO bitacora VALUES ('CREACION DE UN NUEVO USUARIO');  
    RETURN NEW;  
END $BODY$  
LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION fn_after_delete_user()  
RETURNS TRIGGER AS $BODY$  
BEGIN  
    INSERT INTO bitacora VALUES ('ELIMINACIÓN DE UN USUARIO');  
    RETURN NEW;  
END $BODY$  
LANGUAGE plpgsql;
```

```
CREATE TRIGGER tg_after_user  
AFTER INSERT ON users  
FOR EACH ROW  
EXECUTE PROCEDURE fn_after_insert_user();  
  
CREATE TRIGGER tg_after_del_user
```

```
AFTER DELETE ON users
FOR EACH ROW
EXECUTE PROCEDURE fn_after_delete_user();
-- FIN TRIGGER FUNCTIONS

-- SELECTS DE PRUEBA
SELECT * FROM users;
SELECT * FROM bitacora;
SELECT * FROM cursos;
```