

### Evidencias de la ejecución de scripts para el taller 2

1. A continuación, se muestran las tomas de pantalla de la ejecución de los scripts realizados para el primer punto del taller sobre la creación de la tabla de auditoria para las tabla de Mascotas.
  - Muestra de evidencia de la creación de la tabla de MascotasAuditoria.

Name	Value
Updated Rows	0
Query	<pre>CREATE TABLE MASCOTAAUDITORIA(   codigoAuditoria SERIAL UNIQUE,   codigoMascota VARCHAR(10) NOT NULL,   documentoUsuario VARCHAR(15) NOT NULL,   nombreUsuario VARCHAR(100) NOT NULL,   nombreMascota VARCHAR(30) NOT NULL,   edad INT NOT NULL,   especie VARCHAR(30) NOT NULL,   sexo VARCHAR(2) NOT NULL,   tamano NUMERIC NOT NULL,   peligroso BIT NOT NULL,   foto VARCHAR(50) NOT NULL,   activo BIT NOT NULL,   accionAuditoria VARCHAR(15) NOT NULL,   fechaAuditoria TIMESTAMP DEFAULT NOW() NOT NULL )</pre>
Finish time	Sun Sep 12 16:09:11 COT 2021

- Muestra de evidencia de la creación de la función para el trigger.

Name	Value
Updated Rows	0
Query	<pre>CREATE OR REPLACE FUNCTION fcnAuditoriaMascota()   RETURNS TRIGGER   LANGUAGE PLPGSQL   AS   \$\$ DECLARE     varNombreUsuario VARCHAR(100);   BEGIN     SELECT  NOMBREUSUARIO     INTO   varNombreUsuario     FROM   USUARIO     WHERE  DOCUMENTOUSUARIO = CASE WHEN TG_OP = 'INSERT' THEN NEW.DOCUMENTOUSUARIO ELSE OLD.DOCUMENTOUSUARIO END;     IF(TG_OP = 'DELETE') THEN       RAISE 'Accion no permitida';     END IF;     IF EXISTS(SELECT CODIGOMASCOTA FROM MASCOTA WHERE CODIGOMASCOTA = NEW.CODIGOMASCOTA AND TG_OP = 'INSERT') THEN       INSERT INTO MASCOTAAUDITORIA         (CODIGOMASCOTA, DOCUMENTOUSUARIO, NOMBREUSUARIO, NOMBREMASCOTA, EDAD, ESPECIE, SEXO, TAMANO, PELIGROSO, FOTO,         ACTIVO, ACCIONAUDITORIA, FECHAAUDITORIA)       VALUES         (NEW.CODIGOMASCOTA, NEW.DOCUMENTOUSUARIO, varNombreUsuario, NEW.NOMBREMASCOTA, NEW.EDAD, NEW.ESPECIE, NEW.SEXO,         NEW.TAMANO, NEW.PELIGROSO, NEW.FOTO, NEW.ACTIVO, TG_OP, NOW());     RETURN NEW;     ELSEIF (TG_OP = 'UPDATE') THEN       INSERT INTO MASCOTAAUDITORIA         (CODIGOMASCOTA, DOCUMENTOUSUARIO, NOMBREUSUARIO, NOMBREMASCOTA, EDAD, ESPECIE, SEXO, TAMANO, PELIGROSO, FOTO,</pre>

- Evidencia de la creación del trigger.

Name	Value
Updated Rows	0
Query	<pre> /**  * Creación del trigger.  */ DROP TRIGGER IF EXISTS trgAuditoriaMascota ON Mascota; CREATE TRIGGER trgAuditoriaMascota AFTER INSERT OR UPDATE OR DELETE ON Mascota FOR EACH ROW EXECUTE PROCEDURE fcnAuditoriaMascota() </pre>
Finish time	Sun Sep 12 15:44:50 COT 2021

Persistencia de los datos en MascotaAuditoria.

		ASC codigon	ASC docum	ASC nomb	ASC nomb	123 edad	ASC especie	ASC sexo	123 tamano	123 peligroso	ASC foto	ASC acc	fechaauditoria
1	2	MAS004	123456789	Daniela Prado Chaparro	Firulaia 2	3	Perro	M	85,7	1	aqui va la ruta de la foto	1	UPDATE 2021-09-12 16:17:33
2	3	MAS004	123456789	Daniela Prado Chaparro	Firulaia 2	3	Perro	F	85,7	1	aqui va la ruta de la foto	1	UPDATE 2021-09-12 16:17:59
3	4	MAS005	123456789	Daniela Prado Chaparro	Mascota 4	8	Hamster	F	10,5	0	aqui va la ruta de la foto	1	INSERT 2021-09-12 16:44:53
4	5	MAS005	123456789	Daniela Prado Chaparro	Mascota 4	8	Hamster	F	10,5	0	aqui va la ruta de la foto	1	UPDATE 2021-09-12 16:45:41
5	6	MAS005	123456789	Daniela Prado Chaparro	Hams 1	8	Hamster	M	10,7	0	aqui va la ruta de la foto	1	UPDATE 2021-09-12 17:14:19

2. Toma de evidencia para el segundo punto del taller **“Verificación de unicidad de implantación de microchip y esterilización”**

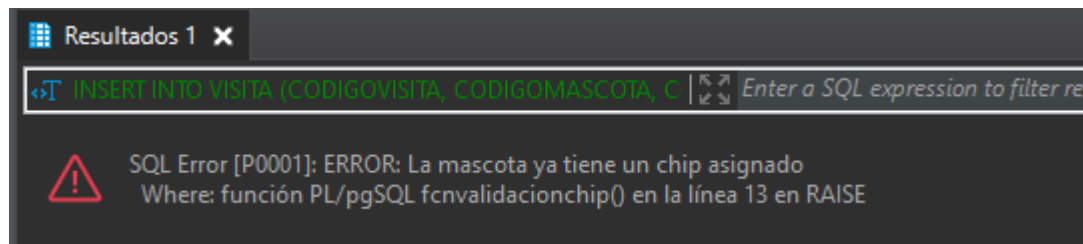
- Creación de la función.

Name	Value
Updated Rows	0
Query	<pre>CREATE OR REPLACE FUNCTION fcnValidacionChip()   RETURNS TRIGGER   LANGUAGE PLPGSQL   AS   \$\$ DECLARE     varChip VARCHAR(50);   BEGIN     SELECT  CHIP     INTO   varChip     FROM   MASCOTA     WHERE  CODIGOMASCOTA = NEW.CODIGOMASCOTA;     IF (NEW.CODITOTIPOVISITA = 'TPV003' AND COALESCE(varChip, 'N') = 'N') THEN       RETURN NEW;     ELSE       RAISE 'La mascota ya tiene un chip asignado';     END IF;   END;   \$\$</pre>
Finish time	Sun Sep 12 17:15:48 COT 2021

- Creación del trigger.

Name	Value
Updated Rows	0
Query	<pre>DROP TRIGGER IF EXISTS trgValidacionChip ON Visita; CREATE TRIGGER trgValidacionChip   BEFORE INSERT ON Visita   FOR EACH ROW   EXECUTE PROCEDURE fcnValidacionChip()</pre>
Finish time	Sun Sep 12 17:01:40 COT 2021

Ejecución del trigger.



### 3. Toma de evidencias para el tercer punto del taller, “Cumplimiento de la meta de implantaciones de microchip”

- Creación de la tabla que persiste la información

Name	Value
Updated Rows	0
Query	<pre> /**  * Creación de la tabla que obtiene las metricas de implantación de chips  */ CREATE TABLE IF NOT EXISTS MicrochipGoals(   fechaInicial DATE NOT NULL,   fechaFinal DATE NOT NULL,   cantidad INT NOT NULL,   cumplimiento VARCHAR(10) NOT NULL ) </pre>
Finish time	Sun Sep 12 19:24:44 COT 2021

- Creación del procedimiento almacenado

Updated Rows	0
Query	<pre> CREATE OR REPLACE PROCEDURE spCumplimientoChips() LANGUAGE PLPGSQL AS \$\$ BEGIN   INSERT INTO MICROCHIPGOALS     (CANTIDAD, CUMPLIMIENTO, FECHAINICIAL, FECHAFINAL)   SELECT  COUNT(CODIGOVISITA) AS cantVisitas, CONCAT((COUNT(CODIGOVISITA) * 100) / 100, '%') AS cumplimiento,     NOW()::DATE-EXTRACT(DOW FROM NOW())::INTEGER-7 AS fechaInicial, NOW()::DATE-EXTRACT(DOW from NOW())::INTEGER AS fechaFinal   FROM    VISITA   WHERE   CODITOTIPOVISITA = 'TPV003'   AND     FECHAREGISTRO BETWEEN NOW()::DATE-EXTRACT(DOW FROM NOW())::INTEGER-7   AND     NOW()::DATE-EXTRACT(DOW from NOW())::INTEGER; END; \$\$ </pre>
Finish time	Sun Sep 12 19:32:32 COT 2021

Ejecución del procedimiento almacenado.

Name	Value
Updated Rows	0
Query	CALL spCumplimientoChips()
Finish time	Tue Sep 14 18:31:29 COT 2021

Persistencia de los datos

SELECT * FROM MICROCHIPGOALS   Enter a SQL expression to filter results (use Ctrl+Space)					
		fechainicial	fechafinal	123 cantidad	ABC cumplimiento
1		2021-09-05	2021-09-12	3	3%
2		2021-09-05	2021-09-12	3	3%
3		2021-09-05	2021-09-12	3	3%