



	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica	Generación de código
	Autor: Juan Jesús Padrón Hernández	alu0100976731
Versión: x	Tiempo invertido:	Fecha : 01/02/19

GENERACIÓN DE CÓDIGO DE COMPONENTES Y PROCEDIMIENTOS (CSI-2)

CÓDIGO DE COMPONENTES
<pre> CREATE TABLE IF NOT EXISTS Musico (dni VARCHAR(9) NOT NULL, Nombre VARCHAR(45) NULL, Nacionalidad VARCHAR(45) NULL, Fecha_nacimiento DATE NULL, PRIMARY KEY (dni)); CREATE TABLE IF NOT EXISTS Grupo (Nombre VARCHAR(20) NOT NULL, PRIMARY KEY (Nombre)); CREATE TABLE IF NOT EXISTS Instrumento (Nombre VARCHAR(20) NOT NULL, PRIMARY KEY (Nombre)); CREATE TABLE IF NOT EXISTS Canciones (Nombre VARCHAR(20) NOT NULL, Duracion FLOAT NULL, Publicacion DATE NULL, Genero VARCHAR(45) NOT NULL, PRIMARY KEY (Nombre)); CREATE TABLE IF NOT EXISTS Discografica (Codigo INT NOT NULL, Nombre VARCHAR(20) NOT NULL, PRIMARY KEY (Codigo)); CREATE TABLE IF NOT EXISTS Productora (Codigo INT NOT NULL, Nombre VARCHAR(45) NOT NULL, PRIMARY KEY (Codigo)); CREATE DOMAIN tipo_album AS TEXT CHECK(VALUE = 'EP' OR VALUE = 'Sencillo' OR VALUE = 'Albumes'); CREATE DOMAIN formato_album AS TEXT CHECK(VALUE = 'Digital' OR VALUE = 'Fisico'); </pre>

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica	Generación de código
	Autor: Juan Jesús Padrón Hernández	alu0100976731
Versión: x	Tiempo invertido:	Fecha : 01/02/19


```

CREATE TABLE IF NOT EXISTS Album (
  Nombre VARCHAR(30) NOT NULL,
  Publicacion DATE NULL,
  Tipo tipo_album NOT NULL,
  Genero VARCHAR(45) NOT NULL,
  Cantidad_canciones INT NOT NULL,
  Formato formato_album NULL,
  PRIMARY KEY (Nombre));

CREATE TABLE IF NOT EXISTS pertenece_cancion_album (
  cancion VARCHAR(45) NOT NULL,
  album VARCHAR(45) NOT NULL,
  PRIMARY KEY (cancion, album),
  CONSTRAINT cancion
    FOREIGN KEY (cancion)
    REFERENCES canciones (nombre)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT album
    FOREIGN KEY (album)
    REFERENCES album (nombre)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX nombre_c_idx_pert ON pertenece_cancion_album (cancion ASC);

CREATE TABLE IF NOT EXISTS Pertenece_Mus_Gru (
  dni_m VARCHAR(9) NOT NULL,
  Nombre_g VARCHAR(45) NOT NULL,
  Comienzo DATE NOT NULL,
  Fin VARCHAR(45) NULL,
  PRIMARY KEY (dni_m, Nombre_g),
  CONSTRAINT dni_m
    FOREIGN KEY (dni_m)
    REFERENCES Musico (dni)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT Nombre_g
    FOREIGN KEY (Nombre_g)
    REFERENCES Grupo (Nombre)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
CREATE INDEX nombre_g_idx on Pertenece_Mus_Gru (Nombre_g ASC);

```


	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica	Generación de código
	Autor: Juan Jesús Padrón Hernández	alu0100976731
Versión: x	Tiempo invertido:	Fecha : 01/02/19

```

CREATE TABLE IF NOT EXISTS Tocan_Musico_Instrumento (
dni_m VARCHAR(9) NOT NULL,
Nombre_i VARCHAR(45) NOT NULL,
PRIMARY KEY (dni_m, Nombre_i),
CONSTRAINT dni_m
FOREIGN KEY (dni_m)
REFERENCES Musico (dni)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT Nombre_i
FOREIGN KEY (Nombre_i)
REFERENCES Instrumento (Nombre)
ON DELETE NO ACTION
ON UPDATE NO ACTION);
CREATE INDEX nombre_i_idx ON Tocan_Musico_Instrumento(Nombre_i ASC);

CREATE TABLE IF NOT EXISTS Asignan_Mus_Gru_Ins (
dni_m VARCHAR(9) NOT NULL,
Nombre_i VARCHAR(45) NOT NULL,
Nombre_g VARCHAR(45) NOT NULL,
PRIMARY KEY (dni_m, Nombre_i, Nombre_g),
CONSTRAINT dni_m
FOREIGN KEY (dni_m)
REFERENCES Musico (dni)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT Nombre_i
FOREIGN KEY (Nombre_i)
REFERENCES Instrumento (Nombre)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT Nombre_g
FOREIGN KEY (Nombre_g)
REFERENCES Grupo (Nombre)
ON DELETE NO ACTION
ON UPDATE NO ACTION);
CREATE INDEX nombre_i_idx_asig on Asignan_Mus_Gru_Ins (Nombre_i ASC);
CREATE INDEX nombre_g_idx_asig on Asignan_Mus_Gru_Ins (Nombre_g ASC);

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica	Generación de código
	Autor: Juan Jesús Padrón Hernández	alu0100976731
Versión: x	Tiempo invertido:	Fecha : 01/02/19


```

CREATE TABLE IF NOT EXISTS Compone_Mus_Can (
  dni_m VARCHAR(9) NOT NULL,
  Nombre_c VARCHAR(45) NOT NULL,
  PRIMARY KEY (dni_m, Nombre_c),
  CONSTRAINT dni_m
    FOREIGN KEY (dni_m)
    REFERENCES Musico (dni)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT Nombre_c
    FOREIGN KEY (Nombre_c)
    REFERENCES Canciones (Nombre)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX nombre_c_idx_compone ON Compone_Mus_Can (Nombre_c ASC);

CREATE TABLE IF NOT EXISTS Distribuido_Alb_Dis (
  Nombre_a VARCHAR(30) NOT NULL,
  Codigo_d INT NOT NULL,
  PRIMARY KEY (Nombre_a, Codigo_d),
  CONSTRAINT nombre_a
    FOREIGN KEY (Nombre_a)
    REFERENCES Album (Nombre)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT codigo_d
    FOREIGN KEY (Codigo_d)
    REFERENCES Discografica (Codigo)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE INDEX codigo_d_idx ON Distribuido_Alb_Dis (Codigo_d ASC);

CREATE TABLE IF NOT EXISTS Publicar_Alb_Pro (
  Nombre_a VARCHAR(30) NOT NULL,
  Codigo_p INT NOT NULL,
  PRIMARY KEY (Nombre_a, Codigo_p),
  CONSTRAINT nombre_a
    FOREIGN KEY (Nombre_a)
    REFERENCES Album (Nombre)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT codigo_p
    FOREIGN KEY (Codigo_p)
    REFERENCES Productora (Codigo)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica	Generación de código
	Autor: Juan Jesús Padrón Hernández	alu0100976731
Versión: x	Tiempo invertido:	Fecha : 01/02/19

CREATE INDEX codigo_p_idx_publicar on Publicar_Alb_Pro (Codigo_p ASC);

CREATE TABLE IF NOT EXISTS Interpreta_Gru_Alb (

Nombre_g VARCHAR(20) NOT NULL,
 Nombre_a VARCHAR(30) NOT NULL,
 PRIMARY KEY (Nombre_g, Nombre_a),
 CONSTRAINT Nombre_g
 FOREIGN KEY (Nombre_g)
 REFERENCES Grupo (Nombre)
 ON DELETE CASCADE
 ON UPDATE CASCADE,
 CONSTRAINT Nombre_a
 FOREIGN KEY (Nombre_a)
 REFERENCES Album (Nombre)
 ON DELETE CASCADE
 ON UPDATE CASCADE);


CREATE INDEX Nombre_a_idx ON Interpreta_Gru_Alb (Nombre_a ASC);

CÓDIGO DE PROCEDIMIENTOS DE OPERACIÓN Y SEGURIDAD

Disparador para comprobar formato de sencillos

```

CREATE OR REPLACE FUNCTION comprobar_single() RETURNS TRIGGER AS
$comprobar_formato_single$
  BEGIN
    IF NEW.tipo = 'Sencillo' THEN
      IF NEW.formato != 'Digital' THEN
        RAISE EXCEPTION '% es sencillo y solo puede tener formato
digital', NEW.Nombre;
      END IF;
    END IF;
    RETURN NEW;
  END;
$comprobar_formato_single$ LANGUAGE plpgsql;
  
```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica	Generación de código
	Autor: Juan Jesús Padrón Hernández	alu0100976731
Versión: x	Tiempo invertido:	Fecha : 01/02/19

CREATE TRIGGER comprobar_single_album AFTER INSERT ON Album
FOR EACH ROW EXECUTE PROCEDURE comprobar_single();

Disparador para comprobar cantidad de canciones según tipo de álbum

CREATE OR REPLACE FUNCTION comprobar_cantidad_canciones() RETURNS TRIGGER
AS \$comprobar_cantidad_canciones_album\$

BEGIN

IF NEW.cantidad_canciones < 1 THEN

RAISE EXCEPTION 'Un álbum no puede tener menos de una canción.';

ELSIF NEW.tipo = 'Sencillo' THEN

IF NEW.cantidad_canciones > 3 THEN

RAISE EXCEPTION '% es un sencillo, no puede tener más de 3 canciones', NEW.Nombre;

END IF;

ELSIF NEW.tipo = 'EP' THEN

IF NEW.cantidad_canciones > 5 THEN

RAISE EXCEPTION '% es un EP, no puede tener más de 5 canciones', NEW.Nombre;

END IF;

ELSIF NEW.tipo = 'Albúmes' THEN

IF NEW.cantidad_canciones < 6 THEN

RAISE EXCEPTION '% es un álbum (LP), no puede tener menos de 6 canciones', NEW.Nombre;

END IF;

END IF;

RETURN NEW;

END;

\$comprobar_cantidad_canciones_album\$ LANGUAGE plpgsql;

CREATE TRIGGER comprobar_cantidad AFTER INSERT ON Album

FOR EACH ROW EXECUTE PROCEDURE comprobar_cantidad_canciones();

Disparador para comprobar los instrumentos que tocan los músicos

CREATE OR REPLACE FUNCTION comprobar_instrumentos() RETURNS TRIGGER AS
\$comprobar_instrumentos_musicos\$

DECLARE

instrumento VARCHAR(45);

inst CURSOR FOR SELECT nombre_i FROM Toca_Musico_Instrumento where
dni_m = NEW.dni_m;

BEGIN

OPEN inst;


LOOP

FETCH inst INTO instrumento;

EXIT WHEN NOT FOUND;

IF instrumento = NEW.Nombre_i THEN

CLOSE inst;

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica	Generación de código
	Autor: Juan Jesús Padrón Hernández	alu0100976731
Versión: x	Tiempo invertido:	Fecha : 01/02/19

```

RETURN NEW;
END IF;
END LOOP;
CLOSE inst;
RAISE EXCEPTION 'El músico % no toca el instrumento',NEW.dni_m;
END;
$comprobar_instrumentos_musicos$ LANGUAGE plpgsql;

CREATE TRIGGER comprobar_instrumentos AFTER INSERT ON Asignan_Mus_Gru_Ins
FOR EACH ROW EXECUTE PROCEDURE comprobar_instrumentos();

Disparador para comprobar género de las canciones
CREATE OR REPLACE FUNCTION comprobar_genero() RETURNS TRIGGER AS
$comprobar_genero_disco$
  DECLARE
    genero_cancion VARCHAR(45);
    genc CURSOR FOR SELECT genero from canciones where nombre =
NEW.cancion;
    genero_album VARCHAR(45);
    gena CURSOR FOR SELECT genero from Album where nombre = NEW.album;
  BEGIN
    OPEN genc;
    OPEN gena;
    LOOP
      FETCH genc INTO genero_cancion;
      EXIT WHEN NOT FOUND;


      FETCH gena INTO genero_album;
      EXIT WHEN NOT FOUND;

      IF genero_cancion = genero_album THEN
        CLOSE genc;
        CLOSE gena;
        RETURN NEW;
      END IF;
    END LOOP;
    CLOSE genc;
    CLOSE gena;
    RAISE EXCEPTION 'El género de la canción % no es el mismo que el del álbum
%', NEW.cancion, NEW.album;

  END;
$comprobar_genero_disco$ LANGUAGE plpgsql;

CREATE TRIGGER comprobar_genero AFTER INSERT ON pertenece_cancion_album
FOR EACH ROW EXECUTE PROCEDURE comprobar_genero();

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)		BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: Tienda Música https://github.com/alu0100997910/abdd-practicas/tree/master/Musica		Generación de código
	Autor: Juan Jesús Padrón Hernández		alu0100976731
Versión: x		Tiempo invertido:	Fecha : 01/02/19