

## CHECK - Respuestas cuestionario 1er Parcial

### Pregunta 1

---

- **Nombre:** Clave primaria de la tabla actuación
- **Enunciado:** La clave primaria de la tabla `actuacion` debería ser creada de la siguiente manera. Marque la o las opciones que considere correctas.
- **Correctas:**
  - ) `CONSTRAINT pk_actuacion PRIMARY KEY (evento_id, artista_id)`
  - ) `CONSTRAINT pk_actuacion PRIMARY KEY (artista_id, evento_id)`

### Pregunta 2

---

- **Nombre:** Historial de nombres de artista
- **Enunciado:** Un artista cambia su nombre artístico (ej: de *Catupecu Machu* a *Vanthra*). La productora necesita mantener un historial de los nombres anteriores del artista. ¿Qué sentencia permite definir correctamente el registro de este historial?
- **Correcta:**

```
CREATE TABLE artista_historial (
    artista_id INT NOT NULL,
    artista_historial_id INT NOT NULL,
    nombre_artistico VARCHAR(100) NOT NULL,
    Fecha_cambio DATE NOT NULL,
    CONSTRAINT pk_artista_historial_id PRIMARY KEY (artista_id,
    artista_historial_id),
    CONSTRAINT fk_artista_historial_artista FOREIGN KEY (artista_id) REFERENCES
    artista(artista_id)
);
```

## Pregunta 3

---

- **Nombre:** Registro de teloneros
- **Enunciado:** La productora necesita registrar los teloneros que puedan tener cada artista en su actuación en un evento. Cada artista en una actuación puede tener o no teloneros. ¿Qué sentencia implementa correctamente este requerimiento?
- **Correcta:**

```
CREATE TABLE actuacion_telonero (
    evento_id INT NOT NULL,
    artista_id INT NOT NULL,
    actuacion_telonero_id int NOT NULL,
    hora_inicio TIME NOT NULL,
    duracion_minutos INT NOT NULL,
    CONSTRAINT pk_actuacion_telonero PRIMARY KEY (evento_id,
    artista_id,actuacion_telonero_id),
    CONSTRAINT fk_actuacion_telonero_actuacion FOREIGN KEY (evento_id, artista_id)
    REFERENCES actuacion(evento_id, artista_id),
    CONSTRAINT fk_actuacion_telonero_artista FOREIGN KEY (actuacion_telonero_id)
    REFERENCES artista(artista_id);
```

---

## Pregunta 4

---

- **Nombre:** Relación de artista con múltiples géneros
- **Enunciado:** La productora quiere registrar que un artista puede pertenecer a varios géneros a la vez (ej: *Rock en Español, New Wave*). ¿Qué cambios harías en el modelo relacional para soportar esta nueva funcionalidad?
- **Correcta:**

```
CREATE TABLE artista_genero
(
    artista_id INT NOT NULL,
    genero_id  INT NOT NULL,
    CONSTRAINT pk_artista_genero PRIMARY KEY (artista_id, genero_id),
    CONSTRAINT fk_artista_genero_genero FOREIGN KEY (genero_id) REFERENCES
    genero_musical(genero_id),
    CONSTRAINT fk_artista_genero_artista FOREIGN KEY (artista_id) REFERENCES
    artista(artista_id)
);
ALTER TABLE artista
DROP fk_artista_genero;
ALTER TABLE artista
DROP COLUMN genero_id;
```

## Pregunta 5

- **Nombre:** Definición de la tabla evento
- **Enunciado:** ¿Qué conjunto de sentencias define correctamente toda la tabla **evento**, según las condiciones establecidas en el problema?
- **Correcta:**

```
CREATE TABLE evento (
    evento_id      INT          NOT NULL,
    nombre         VARCHAR(200)  NOT NULL,
    fecha_inicio   DATE        NOT NULL ,
    cantidad_vendida INT        NOT NULL,
    tipo_evento_id INT        NOT NULL,
    lugar_id       INT        NOT NULL,
    CONSTRAINT pk_evento PRIMARY KEY (evento_id),
    CONSTRAINT fk_evento_tipo_evento FOREIGN KEY (tipo_evento_id) REFERENCES
    tipo_evento(tipo_evento_id),
    CONSTRAINT fk_evento_lugar FOREIGN KEY (lugar_id) REFERENCES lugar(lugar_id),
    CONSTRAINT ck_evento_cantidad_vendida CHECK (cantidad_vendida > 0),
    CONSTRAINT df_evento_fecha_inicio DEFAULT (GETDATE()) FOR fecha_inicio);
```

## Pregunta 6

---

- **Nombre:** Inserciones válidas en continente y país
- **Enunciado:** Dado el modelo relacional, con todas las tablas vacías. Indique qué grupo de sentencias podrá ejecutarse sin problemas.
- **Correcta:** (se evalúan las restricciones no los valores de los datos)

```
Insert into continente (continente_id, nombre) values (1, 'America')
Insert into continente (continente_id, nombre) values (2, 'Europa')

Insert into pais (pais_id, nombre, continente_id) values (1, 'Argentina', 1)
Insert into pais (pais_id, nombre, continente_id) values (2, 'Paraguay', 2)
```

## Pregunta 7

---

- **Nombre:** Inserciones válidas en evento
- **Enunciado:** Dado el modelo relacional, con datos en `lugar` (1: Luna Park, 2: Cocomarola) y en `tipo_evento` (1: Concierto, 2: Festival), ¿qué grupo de sentencias podrá ejecutarse sin problemas en la tabla `evento`?
- **Correcta:**

```
INSERT INTO evento (evento_id, nombre, fecha_inicio, cantidad_vendida,
tipo_evento_id, lugar_id)
VALUES (1, 'Concierto de Rock Nacional 2024', '20241010', 5000, 1, 1);
INSERT INTO evento (evento_id, nombre, cantidad_vendida, tipo_evento_id, lugar_id)
VALUES (2, 'Concierto de Rock Nacional 2025', 10999, 1, 1);
```

## Pregunta 8

---

- **Nombre:** Eliminación de tablas dependientes (evento y lugar)
- **Enunciado:** Dado el modelo relacional con todas las tablas vacías, ¿qué sentencias permiten borrar las tablas indicadas sin violar restricciones de FK?
- **Correcta:** (había dos preguntas iguales correctas - error nuestro- las dos valen)

```
BEGIN TRAN
    ALTER TABLE actuacion
        DROP CONSTRAINT fk_actuacion_evento;
    DROP TABLE evento;
    DROP TABLE lugar;
    DROP TABLE tipo_evento;
    DROP TABLE categoria_lugar;
    DROP TABLE tipo_lugar;
ROLLBACK;
```

## Pregunta 9

---

- **Nombre:** Eliminación de tablas dependientes (artista y país)
- **Enunciado:** Dado el modelo relacional con todas las tablas vacías, ¿qué conjunto de sentencias SQL permitiría borrar las tablas `artista`, `pais`, `continente`, `genero_musical` sin violar restricciones?
- **Correcta:** (en el cuestionario por error, muestra otras opciones como correctas)
  - opción correcta: ) **Todas las opciones**

## Pregunta 10

---

- **Nombre:** Clave incremental en tabla actuacion
- **Enunciado:** Se decide que la tabla `actuacion` tenga una clave primaria simple mediante un ID autoincremental en lugar de una PK compuesta. El cambio debe hacerse sobre una tabla vacía y garantizar la integridad original (un artista solo una vez por evento).
- **Correcta:**

```
ALTER TABLE actuacion
    DROP CONSTRAINT pk_actuacion;
ALTER TABLE actuacion
    ADD actuacion_id INT NOT NULL IDENTITY;
ALTER TABLE actuacion
    ADD CONSTRAINT pk_actuacion PRIMARY KEY (actuacion_id);
ALTER TABLE actuacion
ADD CONSTRAINT uq_actuacion_artista UNIQUE (evento_id, artista_id);
```