



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERÍA

BASES DE DATOS

PROYECTO FINAL BASES DE DATOS: CASO DE ESTUDIO CINE

INTEGRANTES:

No.Cuenta: 319180633 CRUZ MALDONADO ARMANDO

No. Cuenta: 319129456 LÓPEZ REYES ALAM

No.Cuenta: 319049792 MARTÍNEZ PÉREZ BRIAN ERIK

28 DE MAYO DEL 2024

INDICE

CASO DE ESTUDIO “CINE”	2
Diseño Conceptual (MER en DIA)	4
Diseño Lógico	6
Modelo Relacional.....	6
Diccionario de datos.....	6
Normalización	12
TABLA: CINE.....	12
TABLA: CONTENIDO.....	12
TABLA: CULTURAL.....	12
TABLA: DEPORTIVO.....	12
TABLA: DIRECTO.....	13
TABLA: EVENTO.....	13
TABLA: EXHIBE.....	13
TABLA: GENERO, IDIOMA, INGRESO, INTERPRETE, PELICULA, SALA, USUARIO, VISTA.....	13
Diseño Físico	14
DCL.....	14
DDL (Crear Tablas).....	15
DML.....	21
Informes.....	23
CargaInicial.....	25
ValidaTriggers.....	32
Codigo Informes.....	33
Código Carga Inicial.....	34
Tabla.....	39

CASO DE ESTUDIO “CINE”

Una empresa dedicada a la comercialización y mantenimiento de proyectores de cine requiere de una aplicación informática para la difusión de los contenidos que se vayan a exhibir en los cines en los que distribuye su producto; además de proporcionar una plataforma a dichos cines para la gestión de los contenidos que se vayan a emitir en sus salas.

De este modo, es necesario crear un Sistema de Información que gestione los datos de todos los cines a los que la empresa distribuye sus proyectores, así como la programación que en ellos se exhiba.

Para ello, se necesita almacenar toda la información relativa a los cines: identificador, cif (código de identificación fiscal), nombre, teléfono, dirección, logo corporativo, así como su latitud y su longitud para geolocalizarlos en el mapa. Además, se deberá tener en cuenta que un cine podrá disponer de una o más

salas y que de éstas será necesario conocer su nombre, capacidad, hora de apertura de la sala, tiempo de descanso entre sesiones y si dispone de soporte digital o 3D.

En dichas salas se exhibirán una serie de contenidos que podrán ser retransmisiones en directo o películas. De dichos contenidos se desea almacenar: identificador, título, director, duración, año, edad recomendada, género al que pertenece, país de producción, distribuidora, si el contenido se emite en versión original, si se emite en digital o en 3D, el cartel anunciador, el trailer y su web oficial, una breve descripción del contenido, la fecha a partir de la cual se podrá emitir dicho contenido y si el mismo es o no en directo. Los contenidos en directo podrán ser culturales: conciertos, óperas, obras de teatro, ballets; o acontecimientos deportivos. En el caso de los primeros se almacenará el nombre del compositor y el de los intérpretes además del lugar desde dónde se emite; en el caso de los deportes tan sólo se guardará el lugar de origen y el deporte a exhibir. Por otro lado, tendremos las películas que se exhibirán en los cines de las que tan sólo guardaremos su 'imdb'.

Los contenidos programados en la pantalla de cada sala se denominarán eventos y de ellos se almacenarán su identificador, el cine en el que se programan, la sala en cuestión, el contenido programado, la fecha y hora de inicio y quién lo programó. No puede haber eventos programados para una fecha anterior al estreno del contenido. Del mismo modo, no puede haber dos eventos que se solapen en el espacio y en el tiempo.

Las tareas a realizar por la aplicación serán las siguientes: carga inicial, de datos de las tablas ciudad, estado, municipio, género y distribuidor; alta y modificación de contenidos; alta y modificación de cines y salas; programación de eventos en los cines. También se necesita poder publicar la cartelería de un determinado cine de modo que se obtenga la relación de las películas a exhibir.

Requisitos de procesos

P1: Alta de un nuevo cine. El usuario podrá dar de alta nuevos cines introduciendo todos los datos necesarios para la creación de un nuevo cine.

P2: Modificación de un cine. El usuario tendrá la capacidad de modificar los datos referentes a un cine.

P3: Alta de una sala. El usuario podrá introducir los datos necesarios para la creación de una sala que estará ligada al cine.

P4: Modificación de una sala. El usuario podrá modificar los datos correspondientes a una sala. Esto puede ser debido al cambio en el soporte audiovisual de la sala, la capacidad de la misma o la hora de apertura de dicha sala.

P5: Introducción de un nuevo contenido. El usuario introducirá los datos correspondientes al contenido en cuestión (ya sea película o directo).

P6: Modificación de contenidos. La información acerca de cualquier contenido podrá ser modificada.

P7: Añadir un nuevo evento. El usuario podrá incluir contenidos dentro del calendario de exhibición de una determinada sala.

P8: Modificar un evento. El usuario tendrá la capacidad de modificar la exhibición de un evento.

P9: Gestión de la cartelera. El usuario podrá gestionar la cartelera de uno o todos los cines con el fin de publicar los horarios finales de los contenidos de dicho(s) cine(s) hasta una determinada fecha.

P10: Consulta de cartelera. El usuario podrá consultar qué contenidos se exhiben en un determinado cine, dónde hacen un contenido determinado y los horarios de los cines.

P11: Carga inicial: Al iniciar el sistema se realizará una carga de datos actuales.

Listado de restricciones:

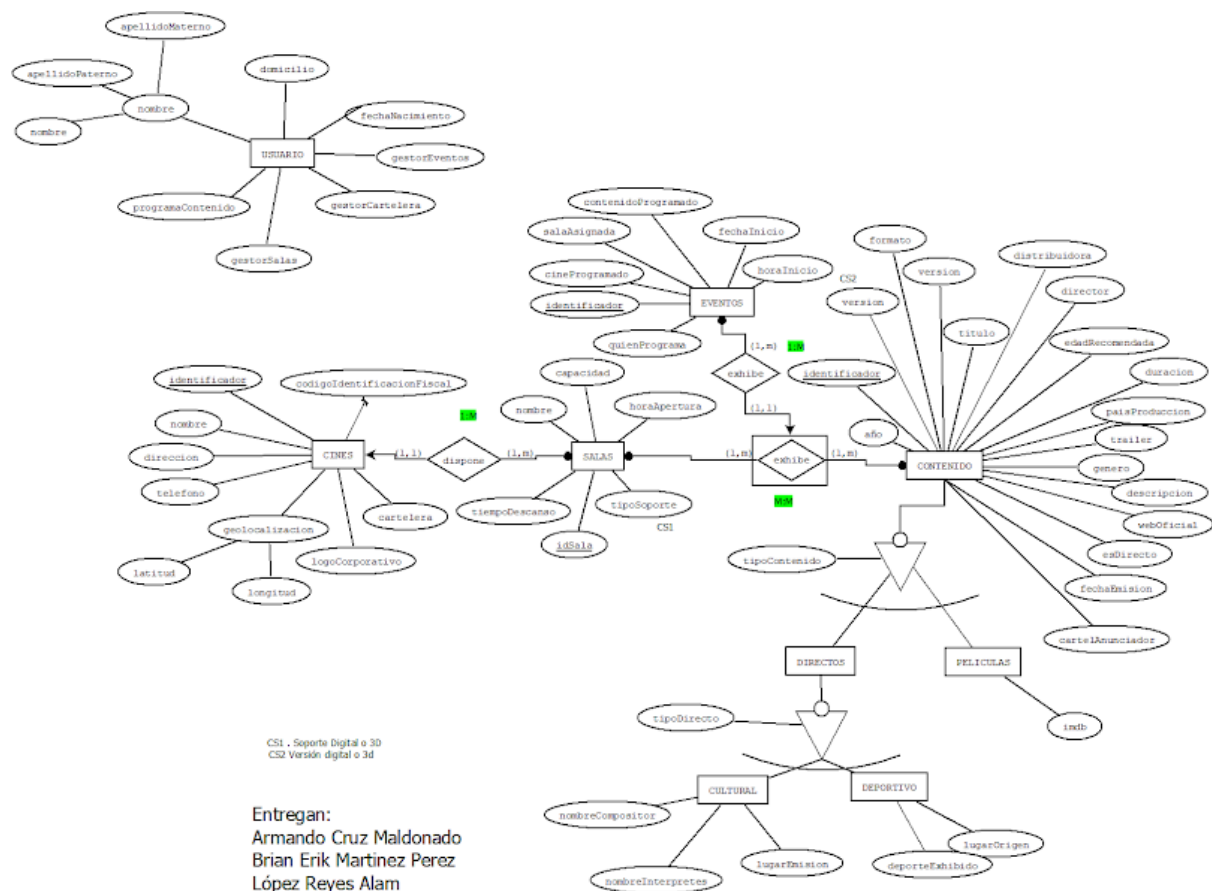
RN1: No pueden programarse dos eventos en la misma sala a la misma hora.

RN2: No se puede modificar un evento si ya ha sido publicado en la cartelera.

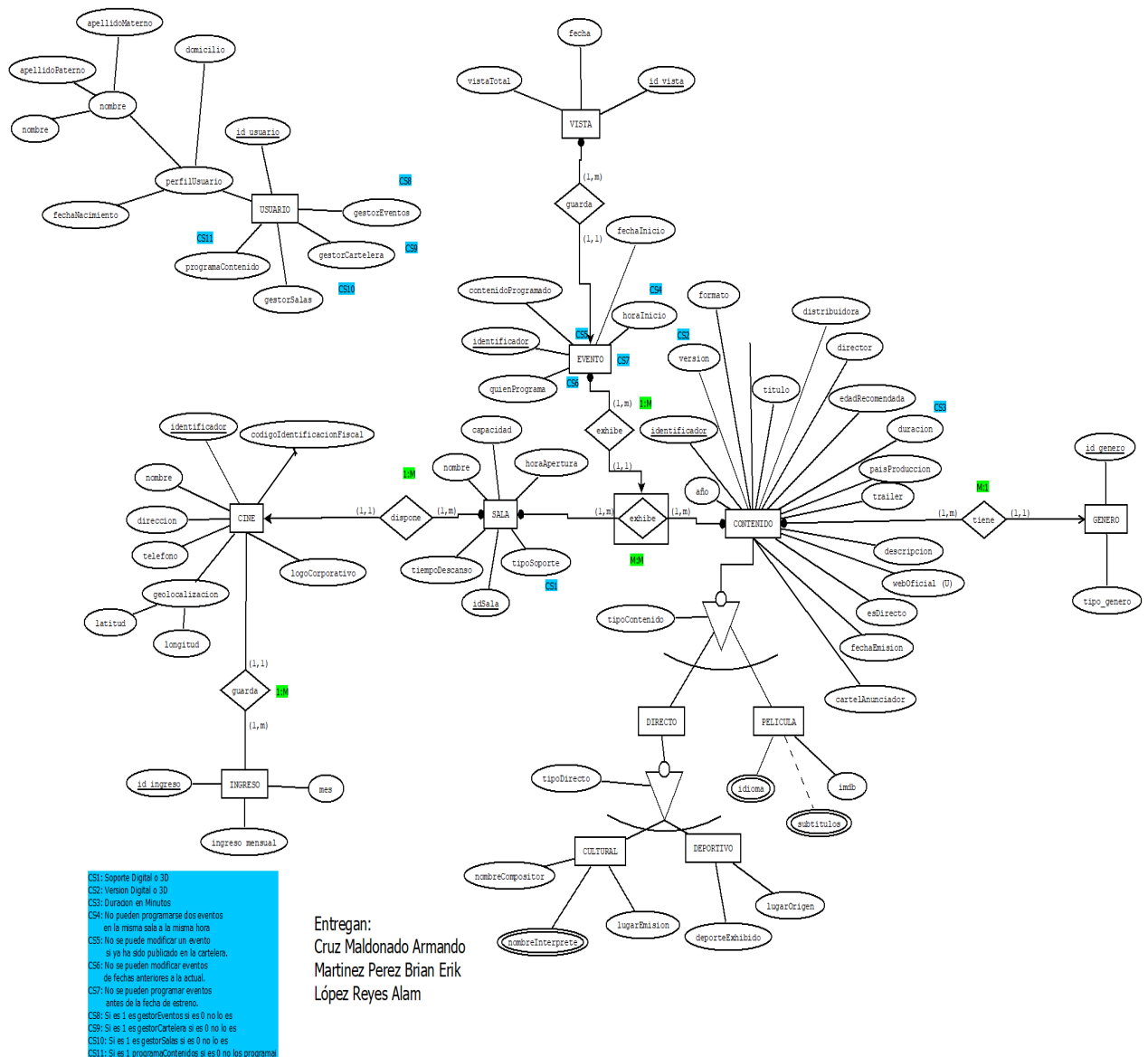
RN3: No se pueden modificar eventos de fechas anteriores a la actual.

RN4: No se pueden programar eventos antes de la fecha de estreno.

Diseño Conceptual (MER en DIA)



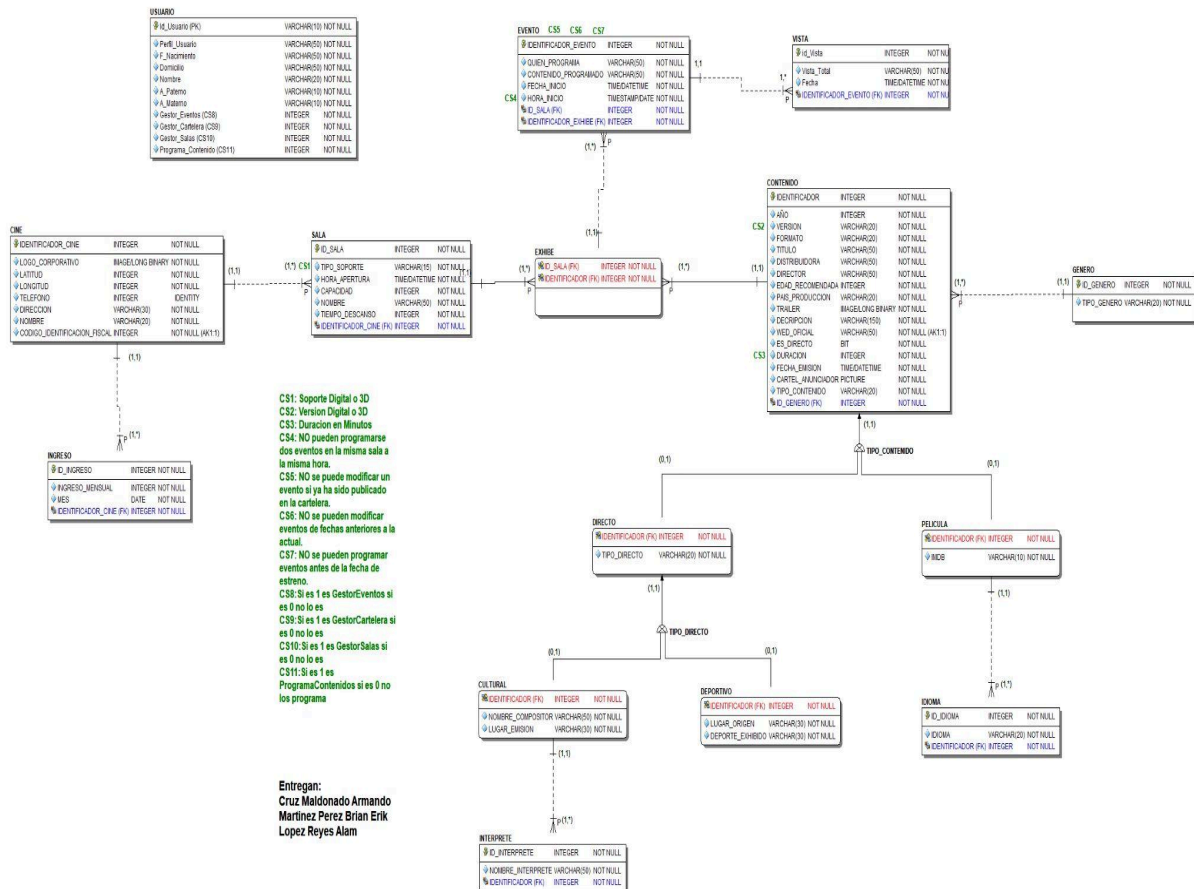
1. Versión 1 del MER en DIA



2. MER corregido en DIA

Diseño Lógico

Modelo Relacional



1. Modelo relacional del caso de estudio

Diccionario de datos

Nombre del objeto: CINE

- **Tipo:** Tabla
- **Descripción:** Almacena información de los cines.
- **Atributos:**
 - IDENTIFICADOR_CINE: int
 - LOGO_CORPORATIVO: varbinary(max)
 - LATITUD: decimal
 - LONGITUD: float
 - TELEFONO: varchar(10)
 - DIRECCION: varchar(30)
 - NOMBRE: varchar(20)
 - CODIGO_IDENTIFICACION_FISCAL: varchar(20)

Nombre del objeto: CONTENIDO

- **Tipo:** Tabla
- **Descripción:** Almacena información del contenido proyectado, como películas y otros eventos.
- **Atributos:**
 - IDENTIFICADOR: int
 - AÑO: int
 - VERSION: varchar(20)
 - FORMATO: varchar(20)
 - TITULO: varchar(50)
 - DISTRIBUIDORA: varchar(50)
 - DIRECTOR: varchar(50)
 - EDAD_RECOMENDADA: int
 - PAIS_PRODUCCION: varchar(20)
 - TRAILER: varbinary(max)
 - DESCRIPCION: varchar(150)
 - WEB_OFICIAL: varchar(255)
 - ES_DIRECTO: bit
 - DURACION: int
 - FECHA_EMISION: date
 - CARTEL_ANUNCIADOR: image
 - TIPO_CONTENIDO: varchar(20)
 - ID_GENERO: int

Nombre del objeto: CULTURAL

- **Tipo:** Tabla
- **Descripción:** Almacena información sobre eventos culturales.
- **Atributos:**
 - IDENTIFICADOR: int
 - NOMBRE_COMPOSITOR: varchar(50)
 - LUGAR_EMISION: varchar(30)

Nombre del objeto: DEPORTIVO

- **Tipo:** Tabla
- **Descripción:** Almacena información sobre eventos deportivos.
- **Atributos:**
 - IDENTIFICADOR: int
 - LUGAR_ORIGEN: varchar(30)
 - DEPORTE_EXHIBIDO: varchar(30)

Nombre del objeto: DIRECTO

- **Tipo:** Tabla
- **Descripción:** Almacena tipos de eventos transmitidos en directo.
- **Atributos:**
 - IDENTIFICADOR: int
 - TIPO_DIRECTO: varchar(20)

Nombre del objeto: EVENTO

- **Tipo:** Tabla
- **Descripción:** Almacena información de eventos programados en el cine.
- **Atributos:**
 - IDENTIFICADOR_EVENTO: int
 - QUIEN_PROGRAMA: varchar(50)
 - CONTENIDO_PROGRAMADO: int
 - FECHA_HORA_INICIO: datetime2(7)
 - ID_SALA: int
 - IDENTIFICADOR_EXHIBE: int
 - VALIDADO: bit

Nombre del objeto: EXHIBE

- **Tipo:** Tabla
- **Descripción:** Relaciona salas con contenido proyectado.
- **Atributos:**
 - ID_SALA: int
 - IDENTIFICADOR: int

Nombre del objeto: GENERO

- **Tipo:** Tabla
- **Descripción:** Almacena géneros de contenido como películas y otros.
- **Atributos:**
 - ID_GENERO: int
 - TIPO_GENERO: varchar(20)

Nombre del objeto: IDIOMA

- **Tipo:** Tabla
- **Descripción:** Almacena idiomas disponibles para el contenido.
- **Atributos:**
 - ID_IDIOMA: int
 - IDIOMA: varchar(40)
 - IDENTIFICADOR: int

Nombre del objeto: INGRESO

- **Tipo:** Tabla
- **Descripción:** Registra ingresos mensuales de cada cine.
- **Atributos:**
 - ID_INGRESO: int
 - INGRESO_MENSUAL: int
 - MES: date
 - IDENTIFICADOR_CINE: int

Nombre del objeto: INTERPRETE

- **Tipo:** Tabla
- **Descripción:** Almacena información sobre los intérpretes de contenido cultural.
- **Atributos:**
 - ID_INTERPRETE: int
 - NOMBRE_INTERPRETE: varchar(50)
 - IDENTIFICADOR: int

Nombre del objeto: PELICULA

- **Tipo:** Tabla
- **Descripción:** Almacena información específica de películas.
- **Atributos:**
 - IDENTIFICADOR: int
 - IMDB: varchar(10)

Nombre del objeto: SALA

- **Tipo:** Tabla
- **Descripción:** Almacena información de las salas de cine.
- **Atributos:**
 - ID_SALA: int
 - TIPO_SOPORTE: varchar(15)
 - HORA_APERTURA: time(7)
 - CAPACIDAD: int
 - NOMBRE: varchar(50)
 - TIEMPO_DESCANSO: int
 - IDENTIFICADOR_CINE: int
 - TIPO_DE_SALA: varchar(15)

Nombre del objeto: USUARIO

- **Tipo:** Tabla
- **Descripción:** Almacena información de usuarios que gestionan diversas actividades del cine.
- **Atributos:**
 - Id_Usuario: varchar(10)
 - Perfil_Usuario: varchar(50)
 - F_Nacimiento: date
 - Domicilio: varchar(255)
 - Nombre: varchar(20)
 - A_Paterno: varchar(10)
 - A_Materno: varchar(10)
 - Gestor_Eventos: bit
 - Gestor_Cartelera: bit
 - Gestor_Salas: bit
 - Programa_Contenido: bit

Nombre del objeto: VISTA

- **Tipo:** Tabla
- **Descripción:** Registra las vistas de eventos.
- **Atributos:**
 - id_Vista: int
 - Vista_Total: int
 - Fecha: date
 - IDENTIFICADOR_EVENTO: int

Indices

Nombre del objeto: IDX_CONTENIDO_ANO_DISTRIBUIDORA

- **Tipo:** Índice
- **Descripción:** Índice no agrupado en los atributos AÑO y DISTRIBUIDORA de la tabla CONTENIDO para mejorar el rendimiento de las consultas que filtran o buscan por año y distribuidora.

Nombre del objeto: IDX_EVENTO_FECHA_HORA_INICIO

- **Tipo:** Índice
- **Descripción:** Índice no agrupado en el atributo FECHA_HORA_INICIO de la tabla EVENTO para soportar eficientemente las consultas por fecha de inicio de eventos, optimizando la búsqueda y el filtrado por fecha.

Nombre del objeto: IDX_INGRESO_MES_CINE

- **Tipo:** Índice
- **Descripción:** Índice agrupado en los atributos MES e IDENTIFICADOR_CINE de la tabla INGRESO para optimizar las operaciones de agregación y reportes de ingresos mensuales por cine, facilitando sumas, promedios y otras agregaciones específicas por rangos de fechas y cines.

Vistas

1. **Nombre del objeto:** vUsuariosPerfiles
 - **Tipo:** Vista
 - **Descripción:** Proporciona una visualización de todos los usuarios, sus cargos y datos personales para administración de roles y usuarios.
2. **Nombre del objeto:** vIngresosCine
 - **Tipo:** Vista
 - **Descripción:** Muestra la relación entre los cines y los ingresos mensuales que generan, permitiendo una gestión financiera efectiva.
3. **Nombre del objeto:** vProgramacionEventos
 - **Tipo:** Vista
 - **Descripción:** Presenta la relación entre eventos programados, las salas donde se muestran y el contenido de dichas salas, facilitando la corroboración de la información mostrada.

4. **Nombre del objeto:** vEventosProgramados
 - **Tipo:** Vista
 - **Descripción:** Ofrece una visión general de los eventos programados disponibles, listando identificadores de eventos, salas y tiempos de inicio.

Triggers

1. **Nombre del objeto:** trg_CheckEventoSolape
 - **Tipo:** Trigger
 - **Descripción:** Asegura que no se programen dos eventos en la misma franja horaria en la misma sala, incluyendo el tiempo de descanso.
2. **Nombre del objeto:** trg_EventoModificacionesBorrado
 - **Tipo:** Trigger
 - **Descripción:** Previene la modificación o eliminación de eventos que han sido validados o cuya fecha ha pasado.

Funciones

1. **Nombre del objeto:** dbo.VistasPorAnio
 - **Tipo:** Función
 - **Descripción:** Calcula y devuelve el total de vistas por año a partir de los datos en la tabla VISTA.
2. **Nombre del objeto:** dbo.GananciasPorCine
 - **Tipo:** Función
 - **Descripción:** Suma y devuelve los ingresos totales por cada cine, utilizando los datos de la tabla INGRESO.

Procedimientos Almacenados

1. **Nombre del objeto:** pusuMontoEntradasPorDiaPorContenido
 - **Tipo:** Procedimiento Almacenado
 - **Descripción:** Consulta el monto de las entradas por día por contenido en cartelera en cada cine.
2. **Nombre del objeto:** pusuTopDiezPeliculasMasVistas
 - **Tipo:** Procedimiento Almacenado
 - **Descripción:** Lista las diez películas más vistas en un rango de fechas especificado.
3. **Nombre del objeto:** pusuIngresosPorCine
 - **Tipo:** Procedimiento Almacenado
 - **Descripción:** Calcula los ingresos por cine en un rango de fechas específico.
4. **Nombre del objeto:** pusuListadoGenerosPorNumeroDeVistas
 - **Tipo:** Procedimiento Almacenado
 - **Descripción:** Genera un listado de géneros por número de vistas, ordenado de mayor a menor.
5. **Nombre del objeto:** pusuListadoUsuariosSistema
 - **Tipo:** Procedimiento Almacenado
 - **Descripción:** Proporciona un listado de todos los usuarios del sistema con sus datos completos.
6. **Nombre del objeto:** pusuInventarioCines

- **Tipo:** Procedimiento Almacenado
- **Descripción:** Muestra un inventario de los cines, incluyendo detalles de las salas.

Normalización

TABLA: CINE

Justificación de normalización:

- **1NF:** Todos los atributos tienen valores atómicos (sin grupos repetitivos).
- **2NF:** No hay dependencias parciales de los atributos no clave sobre la clave primaria.
- **3NF:** No hay dependencias transitivas; todos los atributos no clave dependen directamente de la clave primaria.

No se requiere normalización adicional para esta tabla.

TABLA: CONTENIDO

Justificación de normalización:

- **1NF:** Todos los atributos tienen valores atómicos.
- **2NF y 3NF:** Todos los atributos dependen únicamente de la clave primaria (IDENTIFICADOR) y no existen dependencias entre los atributos no clave.

No se requiere normalización adicional para esta tabla.

TABLA: CULTURAL

Justificación de normalización:

- **1NF:** Todos los atributos son atómicos.
- **2NF y 3NF:** Todos los atributos dependen únicamente de la clave primaria.

No se requiere normalización adicional para esta tabla.

TABLA: DEPORTIVO

Justificación de normalización:

- **1NF:** Todos los atributos son atómicos.
- **2NF y 3NF:** Todos los atributos dependen únicamente de la clave primaria.

No se requiere normalización adicional para esta tabla.

TABLA: DIRECTO

Justificación de normalización:

- **1NF:** Todos los atributos son atómicos.
- **2NF y 3NF:** Todos los atributos dependen únicamente de la clave primaria.

No se requiere normalización adicional para esta tabla.

TABLA: EVENTO

Justificación de normalización:

- **1NF:** Todos los atributos son atómicos.
- **2NF y 3NF:** Todos los atributos dependen únicamente de la clave primaria, excepto por las claves foráneas que están correctamente implementadas.

No se requiere normalización adicional para esta tabla.

TABLA: EXHIBE

Justificación de normalización:

- **1NF:** Todos los atributos son atómicos.
- **2NF y 3NF:** Todos los atributos dependen únicamente de las claves primarias (combinación de ID_SALA y IDENTIFICADOR).

No se requiere normalización adicional para esta tabla.

TABLA: GENERO, IDIOMA, INGRESO, INTERPRETE, PELICULA, SALA, USUARIO, VISTA

Para cada una de estas tablas, la justificación de normalización es similar:

- **1NF:** Todos los atributos son atómicos.
- **2NF:** No hay dependencias parciales.
- **3NF:** No hay dependencias transitivas y todos los atributos dependen directamente de la clave primaria.

Observación adicional: No encontramos problemas evidentes de redundancia o dependencias innecesarias entre atributos en las tablas creadas.

En resumen, las tablas ya están adecuadamente normalizadas hasta la 3NF.

Diseño Físico

DCL

```
13 USE proyectoFinalCine; --0 como se haya nombrado
14 GO
15
16 -- Crear usuarios de ejemplo
17 CREATE LOGIN User1 WITH PASSWORD = 'Password1!';
18 CREATE LOGIN User2 WITH PASSWORD = 'Password2!';
19 CREATE LOGIN User3 WITH PASSWORD = 'Password3!';
20 CREATE LOGIN User4 WITH PASSWORD = 'Password4!';
21 CREATE LOGIN User5 WITH PASSWORD = 'Password5!';
22 CREATE LOGIN User6 WITH PASSWORD = 'Password6!';
23 CREATE LOGIN User7 WITH PASSWORD = 'Password7!';
24 CREATE LOGIN User8 WITH PASSWORD = 'Password8!';
25 CREATE LOGIN User9 WITH PASSWORD = 'Password9!';
26 CREATE LOGIN User10 WITH PASSWORD = 'Password10!';
27 GO
28
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-05-29T04:43:39.7064861-06:00

(creación de los usuarios con contraseñas)

```
41
42 -- Asignar roles y permisos específicos
43 -- Suponemos que algunos usuarios pueden hacer todo, mientras otros tienen restricciones específicas
44
45 -- Usuario 1 tiene acceso a todo
46 EXEC sp_addrolemember 'db_owner', 'User1';
47
48 -- Usuarios para gestión de cines y salas
49 EXEC sp_addrolemember 'db_datawriter', 'User2';
50 EXEC sp_addrolemember 'db_datareader', 'User2';
51
52 EXEC sp_addrolemember 'db_datawriter', 'User3';
53 EXEC sp_addrolemember 'db_datareader', 'User3';
54
55 -- Usuarios para gestión de contenido y eventos
56 EXEC sp_addrolemember 'db_datawriter', 'User4';
57 EXEC sp_addrolemember 'db_datareader', 'User4';
58
59 EXEC sp_addrolemember 'db_datawriter', 'User5';
60 EXEC sp_addrolemember 'db_datareader', 'User5';
61
```

00 %

Messages

Commands completed successfully.

(Asignamos roles específicos)

```
62 -- Usuarios con permisos específicos para consultas
63 GRANT SELECT ON Administrator.CINE TO User6;
64 GRANT SELECT ON Administrator.SALA TO User6;
65
66 GRANT SELECT ON Content.CONTENIDO TO User7;
67 GRANT SELECT ON Content.EVENTO TO User7;
68
69 -- Permiso para actualizar y modificar contenido y eventos
70 GRANT UPDATE, INSERT ON Content.CONTENIDO TO User8;
71 GRANT UPDATE, INSERT ON Content.EVENTO TO User8;
72
73 -- Permiso para gestionar usuarios y roles
74 GRANT ALTER ANY USER TO User9;
75 GRANT ALTER ANY ROLE TO User9;
76
77 -- Usuario 10 es solo para consultas en general
78 GRANT SELECT ON SCHEMA::Administrator TO User10;
79 GRANT SELECT ON SCHEMA::Content TO User10;
80 GO
```

00 %

Messages

Commands completed successfully.

Completion time: 2024-05-29T04:45:42.7697035-06:00

(se otorgan permisos específicos)

DDL (Crear Tablas)

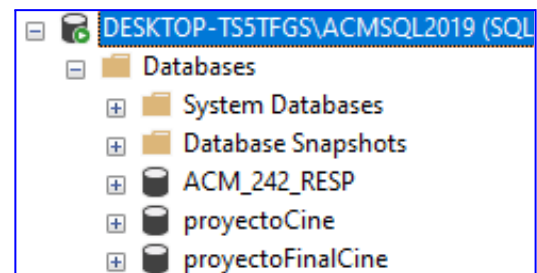
Una vez Teniendo nuestros esquemas, comenzamos con la estructura de la base de datos.

```
-- Crear la base de datos proyectoCine
CREATE DATABASE proyectoFinalCine;
GO
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T02:50:10.6219143-06:00



(Creación de Base de datos ProyectoFinalCine y su resultado en el servidor)

```
-- Crear schemas
CREATE SCHEMA Administrator
GO
CREATE SCHEMA Content
GO
CREATE SCHEMA UserManagment
GO
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T02:59:41.5685480-06:00

(Creación de los
Administrator, Content,

Esquemas
UserManagment)

```
CREATE TABLE Administrator.CINE(
    IDENTIFICADOR_CINE int NOT NULL,
    LOGO_CORPORATIVO varbinary(max) NOT NULL,
    LATITUD decimal NOT NULL,
    LONGITUD float NOT NULL,
    TELEFONO varchar(10) NOT NULL,
    DIRECCION varchar(30) NOT NULL,
    NOMBRE varchar(20) NOT NULL,
    CODIGO_IDENTIFICACION_FISCAL varchar(20) NOT NULL,
    CONSTRAINT CINE_PK PRIMARY KEY NONCLUSTERED (IDENTIFICADOR_CINE),
    CONSTRAINT CHK_TELEFONO CHECK (TELEFONO LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
)
go
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:02:44.2506464-06:00

(Creación de la tabla Administrator.Cine)

```
CREATE TABLE Content.CONTENIDO(
    IDENTIFICADOR int NOT NULL,
    AÑO int NOT NULL,
    VERSION varchar(20) NOT NULL,
    FORMATO varchar(20) NOT NULL,
    TITULO varchar(50) NOT NULL,
    DISTRIBUIDORA varchar(50) NOT NULL,
    DIRECTOR varchar(50) NOT NULL,
    EDAD_RECOMENDADA int NOT NULL,
    PAIS_PRODUCCION varchar(20) NOT NULL,
    TRAILER varbinary(max) NOT NULL,
    DESCRIPCION varchar(150) NOT NULL,
    WEB_OFICIAL varchar(255) NOT NULL,
    ES_DIRECTO bit NOT NULL,
    DURACION int NOT NULL,
    FECHA_EMISION date NOT NULL,
    CARTEL_ANUNCIADOR image NOT NULL,
    TIPO_CONTENIDO varchar(20) NOT NULL,
    ID_GENERO int NOT NULL,
    CONSTRAINT CONTENIDO_PK PRIMARY KEY NONCLUSTERED (IDENTIFICADOR),
    CONSTRAINT CHK_VERSION CHECK (VERSION IN ('Digital', '3D')),
    CONSTRAINT CHK_DURACION CHECK (DURACION > 0),
    CONSTRAINT CHK_DURACION_RANGO CHECK (DURACION BETWEEN 1 AND 480)
)
go
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:07:34.7560073-06:00

(Creación de la tabla content.CONTENIDO)

```
CREATE TABLE Content.CULTURAL(  
    IDENTIFICADOR int NOT NULL,  
    NOMBRE_COMPOSITOR varchar(50) NOT NULL,  
    LUGAR_EMISION varchar(30) NOT NULL,  
    CONSTRAINT CULTURAL_PK PRIMARY KEY NONCLUSTERED (IDENTIFICADOR)  
)  
go
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:11:22.4897876-06:00

(Creación de la tabla content.CULTURAL)

```
CREATE TABLE Content.DEPORTIVO(  
    IDENTIFICADOR int NOT NULL,  
    LUGAR_ORIGEN varchar(30) NOT NULL,  
    DEPORTE_EXHIBIDO varchar(30) NOT NULL,  
    CONSTRAINT DEPORTIVO_PK PRIMARY KEY NONCLUSTERED (IDENTIFICADOR)  
)  
go
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:20:52.6247838-06:00

(Creación de la tabla content.DEPORTIVO)

```
CREATE TABLE Content.DIRECTO(  
    IDENTIFICADOR int NOT NULL,  
    TIPO_DIRECTO varchar(20) NOT NULL,  
    CONSTRAINT DIRECTO_PK PRIMARY KEY NONCLUSTERED (IDENTIFICADOR)  
)  
go
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:24:21.8743795-06:00

(Creación de la tabla content.DIRECTO)

```
CREATE TABLE Content.EVENTO(  
    IDENTIFICADOR_EVENTO int NOT NULL,  
    QUIEN_PROGRAMA varchar(50) NOT NULL,  
    CONTENIDO_PROGRAMADO int NOT NULL,  
    FECHA_HORA_INICIO datetime2(7) NOT NULL,  
    ID_SALA int NOT NULL,  
    IDENTIFICADOR_EXHIBE int NOT NULL,  
    VALIDADO BIT NOT NULL DEFAULT 0,  
    CONSTRAINT EVENTO_PK PRIMARY KEY NONCLUSTERED (IDENTIFICADOR_EVENTO)  
)  
go
```

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:29:46.4518325-06:00

(Creación de la tabla content.EVENTO)

```
CREATE TABLE Content.EXHIBE(  
    ID_SALA          int      NOT NULL,  
    IDENTIFICADOR    int      NOT NULL,  
    CONSTRAINT EXHIBE_PK PRIMARY KEY NONCLUSTERED (ID_SALA, IDENTIFICADOR)  
)  
go
```

0 %

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:31:37.3249735-06:00

(Creación de la tabla content.EXHIBE)

```
CREATE TABLE Content.GENERO(  
    ID_GENERO        int      NOT NULL,  
    TIPO_GENERO      varchar(20) NOT NULL,  
    CONSTRAINT GENERO_PK PRIMARY KEY NONCLUSTERED (ID_GENERO)  
)  
go
```

0 %

Messages

Commands completed successfully.

Completion time: 2024-05-29T03:59:47.9413201-06:00

(Creación de la tabla content.GENERO)

```
CREATE TABLE Content.IDIOMA(  
    ID_IDIOMA        int      NOT NULL,  
    IDIOMA            varchar(40) NOT NULL,  
    IDENTIFICADOR    int      NOT NULL,  
    CONSTRAINT IDIOMA_PK PRIMARY KEY NONCLUSTERED (ID_IDIOMA)  
)  
go
```

0 %

Messages

Commands completed successfully.

Completion time: 2024-05-29T04:09:27.1416864-06:00

(Creación de la tabla content.IDIOMA)

```
CREATE TABLE Administrator.INGRESO(  
    ID_INGRESO        int      NOT NULL,  
    INGRESO_MENSUAL    int      NOT NULL,  
    MES               date     NOT NULL,  
    IDENTIFICADOR_CINE int      NOT NULL,  
    CONSTRAINT INGRESO_PK PRIMARY KEY NONCLUSTERED (ID_INGRESO)  
)  
go
```

0 %

Messages

Commands completed successfully.

Completion time: 2024-05-29T04:11:37.6717256-06:00

(Creación de la tabla Administrator.INGRESO)

```
CREATE TABLE Content.INTERPRETE(  
    ID_INTERPRETE      int          NOT NULL,  
    NOMBRE_INTERPRETE  varchar(50)  NOT NULL,  
    IDENTIFICADOR      int          NOT NULL,  
    CONSTRAINT INTERPRETE_PK PRIMARY KEY NONCLUSTERED (ID_INTERPRETE)  
)  
go
```

%

Messages

Commands completed successfully.

Completion time: 2024-05-29T04:14:45.4582564-06:00

(Creación de la tabla INTERPRETE)

```
CREATE TABLE Content.PELICULA(  
    IDENTIFICADOR      int          NOT NULL,  
    IMDB               varchar(10)  NOT NULL,  
    CONSTRAINT PELICULA_PK PRIMARY KEY NONCLUSTERED (IDENTIFICADOR)  
)  
go
```

```
/*  
 * TABLE: SALA  
 */
```

```
CREATE TABLE Administrator.SALA(  
    ID_SALA            int          NOT NULL,  
    TIPO_SOPORTE       varchar(15)  NOT NULL,  
    HORA_APERTURA     time(7)      NOT NULL,  
    CAPACIDAD          int          NOT NULL,  
    NOMBRE              varchar(50)  NOT NULL,  
    TIEMPO_DESCANSO     int          NOT NULL,  
    IDENTIFICADOR_CINE  int          NOT NULL,  
    TIPO_DE_SALA        varchar(15)  NOT NULL DEFAULT 'Normal',  
    CONSTRAINT SALA_PK PRIMARY KEY NONCLUSTERED (ID_SALA),  
    CONSTRAINT CHK_TIPO_SOPORTE CHECK (TIPO_SOPORTE IN ('Digital', '3D')),  
    CONSTRAINT CHK_CAPACIDAD CHECK (CAPACIDAD BETWEEN 10 AND 1000),  
    CONSTRAINT CHK_TIEMPO_DESCANSO CHECK (TIEMPO_DESCANSO BETWEEN 5 AND 30),  
    CONSTRAINT CHK_TIPO_DE_SALA CHECK (TIPO_DE_SALA IN ('Normal', '2D', '3D', '4D', 'IMAX', 'VIP', 'Junior'))  
)  
go
```

(Creación de la tabla PELICULA y SALA)

```

CREATE TABLE UserManagment.USUARIO(
    Id_Usuario          varchar(10)      NOT NULL,
    Perfil_Usuario      varchar(50)      NOT NULL,
    F_Nacimiento        date              NOT NULL,
    Domicilio          varchar(255)      NOT NULL,
    Nombre              varchar(20)       NOT NULL,
    A_Paterno           varchar(10)       NOT NULL,
    A_Materno           varchar(10)       NOT NULL,
    [Gestor_Eventos]    bit               NOT NULL,
    [Gestor_Cartelera]  bit               NOT NULL,
    [Gestor_Salas]      bit               NOT NULL,
    [Programa_Contenido] bit              NOT NULL,
    CONSTRAINT USUARIO_PK PRIMARY KEY NONCLUSTERED (Id_Usuario),
    CONSTRAINT CHK_Nombre_NoVacio CHECK (Nombre <> '')
)
go

/*
 * TABLE: VISTA
 */

CREATE TABLE Content.VISTA(
    id_Vista            int               NOT NULL,
    Vista_Total         int               NOT NULL,
    Fecha               date              NOT NULL,
    IDENTIFICADOR_EVENTO int              NOT NULL,
    CONSTRAINT VISTA_PK PRIMARY KEY NONCLUSTERED (id_Vista)
)
go

```

(Creación de la tabla USUARIO y VISTA)

```

/*
 * TABLE: CULTURAL
 */

ALTER TABLE Content.CULTURAL ADD CONSTRAINT CULTURAL_DIRECTO_FK
    FOREIGN KEY (IDENTIFICADOR)
    REFERENCES Content.DIRECTO(IDENTIFICADOR)
go

/*
 * TABLE: DEPORTIVO
 */

ALTER TABLE Content.DEPORTIVO ADD CONSTRAINT DEPORTIVO_DIRECTO_FK
    FOREIGN KEY (IDENTIFICADOR)
    REFERENCES Content.DIRECTO(IDENTIFICADOR)
go

/*
 * TABLE: DIRECTO
 */

ALTER TABLE Content.DIRECTO ADD CONSTRAINT DIRECTO_CONTENIDO_FK
    FOREIGN KEY (IDENTIFICADOR)
    REFERENCES Content.CONTENIDO(IDENTIFICADOR)
go

```

(Creación de algunos constraints)

```

-- INDICES
/*
Este índice es apropiado como non-clustered por varias razones:
- La tabla probablemente tiene un índice clustered en la columna IDENTIFICADOR, que es la clave primaria.
- Un índice non-clustered en AÑO y DISTRIBUIDORA mejorará el rendimiento de consultas que filtran o realizan
búsquedas por estos atributos sin alterar el orden físico de almacenamiento de toda la tabla.
*/
CREATE INDEX IDX_CONTENIDO_AÑO_DISTRIBUIDORA ON Content.CONTENIDO(AÑO, DISTRIBUIDORA);

/*
Este índice podría ser tanto clustered como non-clustered dependiendo de los usos específicos, pero aquí se recomienda non-clu
- Si el índice primario ya es clustered en IDENTIFICADOR_EVENTO, entonces un índice non-clustered en FECHA_HORA_INICIO
es ideal para soportar consultas por fecha sin reordenar toda la tabla.
- Un índice non-clustered proporciona acceso rápido a las fechas de los eventos para operaciones de búsqueda y filtrado,
que son comunes en la gestión de eventos.
- Mantener este índice como non-clustered nos permitirá múltiples índices secundarios eficientes junto al índice primario clus
*/
CREATE INDEX IDX_EVENTO_FECHA_HORA_INICIO ON Content.EVENTO(FECHA_HORA_INICIO);

/*
Este índice es un buen candidato para ser clustered por las siguientes razones:
- Un índice clustered en MES e IDENTIFICADOR_CINE reorganiza físicamente los datos en la tabla
según estos atributos, lo que es óptimo para operaciones de agregación y reportes mensuales de ingresos por cine.
- Este ordenamiento facilita operaciones como sumas, promedios y otras agregaciones en rangos de fechas y cines específicos,
haciendo las consultas más rápidas y eficientes en comparación con un índice non-clustered.
*/
CREATE INDEX IDX_INGRESO_MES_CINE ON Administrator.INGRESO(MES, IDENTIFICADOR_CINE);

```

(Creación de índices)

DML

```
/*Explicacion primer vista
En esta primera vista tenemos una visualizacion de todos los usuarios, que cargos ocupan y sus datos personales
Esto con el fin de administrar usuarios y ver los roles que cumplen*/
CREATE VIEW UserManagment.vUsuariosPerfiles AS
SELECT
    Id_Usuario,
    Perfil_Usuario,
    Nombre,
    A_Paterno,
    A_Materno,
    Domicilio,
    F_Nacimiento
FROM
    UserManagment.USUARIO;

select * from UserManagment.vUsuariosPerfiles
go

/*Explicacion segunda vista
Esta es una relacion de los cines con la cantidad de ingresos que reciben este manejo es importante
Porque nos permite administrar financieramente cuanto es el ingreso mensual ademas de entender el rendimiento
que ofrece cada cine*/
CREATE VIEW Administrator.vIngresosCine AS
SELECT
    c.IDENTIFICADOR_CINE AS Cine_ID,
    c.NOMBRE AS Cine_Nombre,
    i.MES,
    i.INGRESO_MENSUAL
FROM
    Administrator.CINE c
JOIN
    Administrator.INGRESO i ON c.IDENTIFICADOR_CINE = i.IDENTIFICADOR_CINE;
```

(Creación de vistas - ya fueron creadas)

```
CREATE TRIGGER trg_CheckEventoSolape
ON Content.EVENTO
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @newIdSala INT, @newFechaHoraInicio DATETIME2, @newDuracion INT, @newFechaHoraFin DATETIME2, @eventoId INT;

    -- Extrae los datos del evento insertado o actualizado junto con la duración del contenido asociado.
    SELECT @newIdSala = i.ID_SALA, @newFechaHoraInicio = i.FECHA_HORA_INICIO, @newDuracion = c.DURACION, @eventoId = i.IDENTIFICADOR_EVENTO
    FROM inserted i
    JOIN Content.CONTENIDO c ON i.CONTENIDO_PROGRAMADO = c.IDENTIFICADOR;

    -- Calcula el tiempo de finalización del evento, teniendo en cuenta la duración del contenido.
    SET @newFechaHoraFin = DATEADD(MINUTE, @newDuracion, @newFechaHoraInicio);

    -- Verifica si hay algún evento existente que se solape con el nuevo evento.
    IF EXISTS (
        SELECT 1
        FROM Content.EVENTO e
        JOIN Administrator.SALA s ON e.ID_SALA = s.ID_SALA
        WHERE e.ID_SALA = @newIdSala
        AND e.IDENTIFICADOR_EVENTO <> @eventoId -- Excluye el evento actual en caso de actualización para evitar auto-conflict
        AND (
            (@newFechaHoraInicio BETWEEN e.FECHA_HORA_INICIO AND DATEADD(MINUTE, @newDuracion + s.TIEMPO_DESCANSO, e.FECHA_HORA_INICIO)
            OR
            (@newFechaHoraFin BETWEEN e.FECHA_HORA_INICIO AND DATEADD(MINUTE, @newDuracion + s.TIEMPO_DESCANSO, e.FECHA_HORA_INICIO)
            OR
            (e.FECHA_HORA_INICIO BETWEEN @newFechaHoraInicio AND @newFechaHoraFin)
        )
    )

```

(Desarrollo de uno de los triggers - ya fue creado)

```

-- FUNCIONES
USE proyectoCine; --0 como se haya nombrado
GO

CREATE FUNCTION dbo.VistasPorAnio()
RETURNS TABLE
AS
RETURN (
    SELECT
        YEAR(Fecha) AS Year,
        SUM(Vista_Total) AS TotalViews
    FROM
        Content.VISTA
    GROUP BY
        YEAR(Fecha)
);

CREATE FUNCTION dbo.GananciasPorCine()
RETURNS TABLE
AS
RETURN (
    SELECT
        IDENTIFICADOR AS CineID,
        SUM(INGRESO_MENSUAL) AS TotalIncome
    FROM
        Administrator.INGRESO
    GROUP BY
        IDENTIFICADOR
);

```

(Funciones creadas)

```

-- 1. Monto de Entradas por Día por Contenido en Cartelera
CREATE PROCEDURE pusuMontoEntradasPorDiaPorContenido
AS
BEGIN
    SELECT C.*, I.Ingreso_Mensual_Mayor
    FROM Administrator.CINE AS C
    LEFT JOIN ( SELECT IDENTIFICADOR_CINE, MAX(INGRESO_MENSUAL) AS Ingreso_Mensual_Mayor
                FROM Administrator.INGRESO
                GROUP BY IDENTIFICADOR_CINE
              ) AS I ON C.IDENTIFICADOR_CINE = I.IDENTIFICADOR_CINE
END
GO

-- 2. Top Diez de las Películas Más Vistas
CREATE PROCEDURE pusuTopDiezPeliculasMasVistas
    @FechaInicio DATE,
    @FechaFin DATE
AS
BEGIN
    SELECT TOP 10
        C.TITULO AS Pelicula,
        SUM(V.Vista_Total) AS Total_Vistas
    FROM Content.VISTA AS V
    INNER JOIN Content.EVENTO AS E ON V.IDENTIFICADOR_EVENTO = E.IDENTIFICADOR_EVENTO

```

(Creación store procedures)

```

326
327 EXEC pusuMontoEntradasPorDiaPorContenido;
328
329 -- requiere dos parámetros de fecha (inicio y fin del periodo a consultar)
330 EXEC pusuTopDiezPelículasMasVistas @FechaInicio = '2023-01-01', @FechaFin = '2023-06-02';
331 --requiere fechas de inicio y fin
332 EXEC pusuIngresosPorCine @FechaInicio = '2024-01-01', @FechaFin = '2024-03-01';
333 EXEC pusuListadoGenerosPorNumeroDeVistas;
334 EXEC pusuListadoUsuariosSistema;
335 EXEC pusuInventarioCines;
336

```

	IDENTIFICADOR_CINE	LOGO_CORPORATIVO	LATITUD	LONGITUD	TELEFONO	DIRECCION	NOMBRE	CODIGO_IDENTIFICACION_FISCAL	Ingreso_Mensual_Me
1	1	0x	19	-99.133209	5555555555	Calle Uno 123	Cine Plaza	CIF001	135000
2	2	0x	19	-99.141266	5555555556	Calle Dos 456	Cine Sol	CIF002	170000
3	3	0x	19	-99.14245	5555555557	Calle Tres 789	Cine Luna	CIF003	155000
4	4	0x	19	-99.137845	5555555558	Calle Cuatro ...	Cine Estr...	CIF004	165000
5	5	0x	19	-99.138916	5555555559	Calle Cinco ...	Cine Co...	CIF005	175000
6	6	0x	19	-99.139987	5555555560	Calle Seis 303	Cine Ecli...	CIF006	NULL
7	7	0x	19	-99.140356	5555555561	Calle Siete 4...	Cine Gal...	CIF007	NULL

(Pruebas de los SP)

Informes

a) Un informe del monto de entradas por día por cada contenido en cartelera. Para ello el área contable asigna un folio consecutivo del 1 al n por cada día que está en cartelera un contenido (fecha, monto de entradas vendidas). Este informe se puede obtener también por cine.

a) Un informe del monto de entradas por día por cada contenido en cartelera. Para ello el área contable asigna un folio consecutivo del 1 al n por cada día que está en cartelera un contenido (fecha, monto de entradas vendida). Este informe se puede obtener también por cine
*/

	IDENTIFICADOR_CINE	LOGO_CORPORATIVO	LATITUD	LONGITUD	TELEFONO	DIRECCION	NOMBRE	CODIGO_IDENTIFICACION_FISCAL	Ingreso_Mensual_Mayor
1	1	0x	19	-99.133209	5555555555	Calle Uno 123	Cine Plaza	CIF001	135000
2	2	0x	19	-99.141266	5555555556	Calle Dos 456	Cine Sol	CIF002	170000
3	3	0x	19	-99.14245	5555555557	Calle Tres 789	Cine Luna	CIF003	155000
4	4	0x	19	-99.137845	5555555558	Calle Cuatro 101	Cine Estrella	CIF004	165000
5	5	0x	19	-99.138916	5555555559	Calle Cinco 202	Cine Cometa	CIF005	175000
6	6	0x	19	-99.139987	5555555560	Calle Seis 303	Cine Eclipse	CIF006	NULL
7	7	0x	19	-99.140356	5555555561	Calle Siete 404	Cine Galaxia	CIF007	NULL
8	8	0x	19	-99.141425	5555555562	Calle Ocho 505	Cine Cosmos	CIF008	NULL
9	9	0x	19	-99.142494	5555555563	Calle Nueve 606	Cine Orion	CIF009	NULL
10	10	0x	19	-99.143563	5555555564	Calle Diez 707	Cine Meteor	CIF010	NULL

Informe A

b) Top ten de las películas más vistas en un periodo de tiempo, incluyendo la cantidad monto.

```

-- b) Top ten de las películas más vistas en un periodo de tiempo, incluyendo la cantidad monto.

```

	Película	Total_Vistas
1	The Lord of the Rings: The Fellowship of the Ring	6000
2	Pulp Fiction	5900
3	Titanic	5800
4	Inception	5700
5	The Dark Knight	5600
6	The Matrix	5500
7	Jurassic Park	5400
8	Star Wars: Episode V - The Empire Strikes Back	5300
9	The Godfather	5200
10	The Shawshank Redemption	5100

Informe B

c) Ingresos por cine (todos los cines, nombre y dirección) en una fecha o rango de fechas.

```
--c) Ingresos por cine (todos los cines, nombre y dirección) en una fecha o rango de fechas
```

	NOMBRE	DIRECCION	INGRESO_MENSUAL	MES
1	Cine Plaza	Calle Uno 123	100000	2024-01-01
2	Cine Sol	Calle Dos 456	150000	2024-01-01
3	Cine Luna	Calle Tres 789	120000	2024-01-01
4	Cine Plaza	Calle Uno 123	130000	2024-02-01
5	Cine Sol	Calle Dos 456	160000	2024-02-01
6	Cine Luna	Calle Tres 789	140000	2024-02-01
7	Cine Plaza	Calle Uno 123	110000	2024-03-01
8	Cine Sol	Calle Dos 456	170000	2024-03-01
9	Cine Luna	Calle Tres 789	115000	2024-03-01
10	Cine Estrella	Calle Cuatro 101	125000	2024-03-01

Informe C

d) Listado de géneros, número de vistas ordenados de mayor a menor

```
--d) Listado de géneros, numero de vistas ordenados de mayor a menor
```

	TIPO_GENERO	Numero_Genero
1	Drama	3
2	Crímen	3
3	Ciencia Ficción	3
4	Aventura	3
5	Musical	3
6	Deportivo	2
7	Acción	2
8	Romance	1

Informe D

e) Listado de los usuarios del sistema, nombre completo, fecha de nacimiento y domicilio

```
--e) Listado de los usuarios del sistema, nombre completo, fecha de nacimiento y domicilio
--π_(NOMBRE + A_PATERNO + A_MATERNO, F_Nacimiento, Domicilio)(USUARIO)
```

	Nombre Completo	F_Nacimiento	Domicilio
1	Juan Pérez García	1980-01-01	123 Calle Principal, Ciudad
2	Ana López Martínez	1990-02-02	456 Calle Secundaria, Ciudad
3	Luis Torres Hernández	1985-03-03	789 Calle Terciana, Ciudad
4	María Ruiz Díaz	1992-04-04	159 Calle Cuarta, Ciudad
5	Carlos Navarro Molina	1983-05-05	753 Calle Quinta, Ciudad
6	Sofía Méndez Prieto	1995-06-06	357 Calle Sexta, Ciudad
7	Roberto Blanco Casas	1988-07-07	951 Calle Séptima, Ciudad
8	Elena Santos Gómez	1979-08-08	123 Calle Octava, Ciudad
9	Tomás Reyes Lozano	1991-09-09	654 Calle Novena, Ciudad
10	Irene García Femán...	1986-10-10	321 Calle Décima, Ciudad

Informe E

f) Inventario de los cines: nombre, dirección, sala, número de asientos, tipo de sala (normal, 2d, 3d, 4d, IMAX, vip, junior)

f) Inventario de los cines: nombre, dirección, sala, número de asientos, tipo de sala (normal, 2d, 3d, 4d, IMAX, vip, junior)
*/
--π_CINE.NOMBRE, CINE.DIRECCION, SALA.NOMBRE, SALA.CAPACIDAD, SALA.TIPO_DE_SAL (CINE ⋈(CINE.IDENTIFICADOR_CINE

	Nombre Cine	DIRECCION	Nombre Sala	número de asientos	TIPO_DE_SALA
1	Cine Plaza	Calle Uno 123	Sala 1	100	Normal
2	Cine Plaza	Calle Uno 123	Sala 2	150	Normal
3	Cine Sol	Calle Dos 456	Sala 1	200	Normal
4	Cine Sol	Calle Dos 456	Sala 2	120	Normal
5	Cine Luna	Calle Tres 789	Sala 1	300	Normal
6	Cine Luna	Calle Tres 789	Sala 2	180	Normal
7	Cine Plaza	Calle Uno 123	Sala 3	250	Normal
8	Cine Sol	Calle Dos 456	Sala 4	220	Normal
9	Cine Luna	Calle Tres 789	Sala 3	110	Normal
10	Cine Plaza	Calle Uno 123	Sala 4	130	Normal

Informe F

Cargalnicial

cargalnicial.sql - DELL_BRIAN\BEMPSQL2019.proyectoCine (DELL_BRIAN\brian (55))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

- Connect
- BEMP_Historial
- lab_24_2_319049792
- MPBE_242_RESP
- proyectoCine
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - Administrator.CINE
 - Administrator.INGRESO
 - Administrator.SALA
 - Content.CONTENIDO
 - Content.CULTURAL
 - Content.DEPORTIVO
 - Content.EVENTO
 - Content.EXHIBE
 - Content.GENERO
 - Content.IDIOMA
 - Content.INTERPRETE
 - Content.PELICULA
 - Content.VISTA
 - UserManagement.USUARIO
 - Views
 - External Resources
 - Synonyms
 - Procedures

Query Editor

DML-24-2 (4 de Abr..._BRIAN\brian (65)) cargalnicial.sql - DELL_BRIAN\brian (55))*

```
GO

--el resultado de un select a cada tabla de la base de datos, incluyendo un count() a cada una de las tablas
select * from Administrator.CINE
select count(*) from Administrator.CINE
```

Results

	IDENTIFICADOR	LOGO_CORPORATIVO	LATITUD	LONGITUD	TELEFONO	DIRECCION	NOMBRE	CODIGO_IDENTIFICACION_FISCAL
1	1	0x	19	-99.133209	5555555555	Calle Uno 123	Cine Plaza	CIF001
2	2	0x	19	-99.141266	5555555556	Calle Dos 456	Cine Sol	CIF002
3	3	0x	19	-99.14245	5555555557	Calle Tres 789	Cine Luna	CIF003
4	4	0x	19	-99.137845	5555555558	Calle Cuatro 101	Cine Estrella	CIF004
5	5	0x	19	-99.138916	5555555559	Calle Cinco 202	Cine Cometa	CIF005
6	6	0x	19	-99.139987	5555555560	Calle Seis 303	Cine Eclipse	CIF006
7	7	0x	19	-99.140356	5555555561	Calle Siete 404	Cine Galaxia	CIF007
8	8	0x	19	-99.141425	5555555562	Calle Ocho 505	Cine Cosmos	CIF008
9	9	0x	19	-99.142494	5555555563	Calle Nueve 606	Cine Orion	CIF009
10	10	0x	19	-99.143563	5555555564	Calle Diez 707	Cine Meteor	CIF010

(No column name)

1	10
---	----

Query executed successfully.

DELL_BRIAN\BEMPSQL2019 (15... DELL_BRIAN\brian (55) proyectoCine | 00:00:00 | 11 rows

(select a la tabla CINE, incluyendo un count() a la tabla CINE)

Microsoft SQL Server Management Studio interface showing a query executed successfully. The query is:

```
select * from Administrator.INGRESO
select count(*) from Administrator.INGRESO
```

The results window displays the following data:

ID_INGRESO	INGRESO_MENSUAL	MES	IDENTIFICADOR
1	100000	2024-01-01	1
2	150000	2024-01-01	2
3	120000	2024-01-01	3
4	130000	2024-02-01	1
5	160000	2024-02-01	2
6	140000	2024-02-01	3
7	110000	2024-03-01	1
8	170000	2024-03-01	2
9	115000	2024-03-01	3
10	125000	2024-03-01	4
11	135000	2024-04-01	1
12	145000	2024-04-01	2
13	155000	2024-04-01	3
14	165000	2024-04-01	4
15	175000	2024-04-01	5

The status bar indicates: Query executed successfully. DELL_BRIAN\BEMPSQL2019 (15... DELL_BRIAN\brian (55)) proyectoCine 00:00:00 16 rows

(select a la tabla INGRESO, incluyendo un count() a la tabla INGRESO)

Microsoft SQL Server Management Studio interface showing a query executed successfully. The query is:

```
select * from Administrator.SALA
select count(*) from Administrator.SALA
```

The results window displays the following data:

ID_SALA	TIPO_SOPORTE	HORA_APERTURA	CAPACIDAD	NOMBRE	TIEMPO_DESCANSO	IDENTIFICADOR
1	Digital	10:00:00.00000000	100	Sala 1	15	1
2	3D	10:00:00.00000000	150	Sala 2	15	1
3	Digital	11:00:00.00000000	200	Sala 1	20	2
4	3D	11:00:00.00000000	120	Sala 2	20	2
5	Digital	12:00:00.00000000	300	Sala 1	10	3
6	3D	12:00:00.00000000	180	Sala 2	10	3
7	Digital	10:30:00.00000000	250	Sala 3	12	1
8	3D	10:30:00.00000000	220	Sala 4	12	2
9	Digital	09:00:00.00000000	110	Sala 3	15	3
10	3D	09:00:00.00000000	130	Sala 4	15	1

The status bar indicates: Query executed successfully. DELL_BRIAN\BEMPSQL2019 (15... DELL_BRIAN\brian (55)) proyectoCine 00:00:00 11 rows

(select a la tabla SALA, incluyendo un count() a la tabla SALA)

Microsoft SQL Server Management Studio interface showing a query execution for the 'CONTENIDO' table. The query is:

```
select count(*) from Content.CONTENIDO
```

The results pane displays a single row with the value 20.

IDENTIFICADOR	AÑO	VERSION	FORMATO	TITULO	DISTRIBUIDORA	DIRECTOR	EDAD_RECOMENDADA
1	1994	Digital	Pelicula	The Shawshank Redemption	Columbia Pictures	Frank Darabont	16
2	1972	Digital	Pelicula	The Godfather	Paramount Pictures	Francis Ford Coppola	18
3	1980	Digital	Pelicula	Star Wars: Episode V - The Empire Strikes Back	20th Century Fox	Irvin Kershner	12
4	1993	Digital	Pelicula	Jurassic Park	Universal Pictures	Steven Spielberg	13
5	1999	Digital	Pelicula	The Matrix	Warner Bros	Lana Wachowski, Lilly Wachowski	15
6	2008	Digital	Pelicula	The Dark Knight	Warner Bros	Christopher Nolan	15
7	2010	Digital	Pelicula	Inception	Warner Bros	Christopher Nolan	13
8	1997	Digital	Pelicula	Titanic	Paramount Pictures	James Cameron	12
9	1994	Digital	Pelicula	Pulp Fiction	Miramax Films	Quentin Tarantino	18
10	2001	Digital	Pelicula	The Lord of the Rings: The Fellowship of the Ring	New Line Cinema	Peter Jackson	12
11	2003	Digital	Pelicula	The Lord of the Rings: The Return of the King	New Line Cinema	Peter Jackson	12

(select a la tabla CONTENIDO, incluyendo un count() a la tabla CONTENIDO)

Microsoft SQL Server Management Studio interface showing a query execution for the 'CULTURAL' table. The query is:

```
select count(*) from Content.CULTURAL
```

The results pane displays a single row with the value 3.

IDENTIFICADOR	NOMBRE_COMPOSITOR	LUGAR_EMISION
16	Coldplay	London, UK
18	William Shakespeare	Stratford-upon-Avon, UK
20	Giuseppe Verdi	New York, USA

(select a la tabla CULTURAL, incluyendo un count() a la tabla CULTURAL)

Microsoft SQL Server Management Studio interface showing a query execution for the 'Content.DEPORTIVO' table.

Query executed:

```
select * from Content.DEPORTIVO
select count(*) from Content.DEPORTIVO
```

Results:

IDENTIFICADOR	LUGAR_ORIGEN	DEPORTE_EXHIBIDO
17	Las Vegas, USA	Chess
19	Los Angeles, USA	Basketball

Summary:

(No column name)
2

Query executed successfully. DELL_BRIAN\BEMPSQL2019 (15... DELL_BRIAN\brian (55)) proyectoCine 00:00:00 3 rows

(select a la tabla DEPORTIVO, incluyendo un count() a la tabla DEPORTIVO)

Microsoft SQL Server Management Studio interface showing a query execution for the 'Content.DIRECTO' table.

Query executed:

```
select count(*) from Content.DIRECTO
select * from Content.DIRECTO
select count(*) from Content.DIRECTO
```

Results:

IDENTIFICADOR	TIPO_DIRECTO
16	Concierto
17	Deporte
18	Teatro
19	Deporte
20	Ópera

Summary:

(No column name)
5

Query executed successfully. DELL_BRIAN\BEMPSQL2019 (15... DELL_BRIAN\brian (55)) proyectoCine 00:00:00 6 rows

(select a la tabla DIRECTO, incluyendo un count() a la tabla DIRECTO)

Microsoft SQL Server Management Studio interface showing a query executed successfully. The query is:

```
select * from Content.EVENTO
select count(*) from Content.EVENTO
```

The results pane displays the following data:

IDENTIFICADOR	QUIEN_PROGRAMA	CONTENIDO_PROGRAMADO	FECHA_HORA_INICIO	ID_SALA	IDENTIFICADOR_EVENTO
10	User5	20	2023-01-02 18:00:00.0000000	5	20
11	User1	16	2023-01-03 10:00:00.0000000	1	16
12	User2	17	2023-01-03 12:00:00.0000000	2	17
13	User3	18	2023-01-03 14:00:00.0000000	3	18
14	User4	19	2023-01-03 16:00:00.0000000	4	19
15	User5	20	2023-01-03 18:00:00.0000000	5	20
16	User1	16	2023-01-04 10:00:00.0000000	1	16
17	User2	17	2023-01-04 12:00:00.0000000	2	17
18	User3	18	2023-01-04 14:00:00.0000000	3	18
19	User4	19	2023-01-04 16:00:00.0000000	4	19
20	User5	20	2023-01-04 18:00:00.0000000	5	20

The count result is 20.

(select a la tabla EVENTO, incluyendo un count() a la tabla EVENTO)

Microsoft SQL Server Management Studio interface showing a query executed successfully. The query is:

```
select * from Content.EXHIBE
select count(*) from Content.EXHIBE
```

The results pane displays the following data:

ID_SALA	IDENTIFICADOR
1	16
2	17
3	18
4	19
5	20

The count result is 5.

(select a la tabla EXHIBE, incluyendo un count() a la tabla EXHIBE)

Microsoft SQL Server Management Studio interface showing a query execution for the 'Content.GENERO' table.

Query executed:

```
select count(*) from Content.GENERO
```

Results:

ID_GENERO	TIPO_GENERO
1	Drama
2	Crimen
3	Acción
4	Ciencia Ficción
5	Romance
6	Aventura
7	Terror
8	Comedia
9	Documental
10	Musical
11	Deportivo

Query executed successfully. 12 rows.

(select a la tabla GÉNERO, incluyendo un count() a la tabla GÉNERO)

Microsoft SQL Server Management Studio interface showing a query execution for the 'Content.IDIOMA' table.

Query executed:

```
select count(*) from Content.IDIOMA
```

Results:

ID_IDIOMA	IDIOMA	IDENTIFICADOR
4	Inglés	4
5	Inglés	5
6	Inglés	6
7	Inglés	7
8	Inglés	8
9	Inglés	9
10	Inglés	10
11	Inglés	11
12	Inglés	12
13	Inglés	13
14	Inglés	14
15	Inglés	15

Query executed successfully. 16 rows.

(select a la tabla IDIOMA, incluyendo un count() a la tabla IDIOMA)

Microsoft SQL Server Management Studio interface showing a query executed successfully. The query is:

```
select * from Content.INTERPRETE
select count(*) from Content.INTERPRETE
```

The results pane displays the following data:

ID_INTERPRETE	NOMBRE_INTERPRETE	IDENTIFICADOR
1	Chris Martin	16
2	Benedict Cumberbatch	18
3	Diana Damrau	20

The status bar indicates: Query executed successfully. DELL_BRIAN\BEMPSQL2019 (15... DELL_BRIAN\brian (55)) proyectoCine | 00:00:00 | 4 rows

(select a la tabla INTERPRETE, incluyendo un count() a la tabla INTERPRETE)

Microsoft SQL Server Management Studio interface showing a query executed successfully. The query is:

```
select * from Content.PELICULA
select count(*) from Content.PELICULA
```

The results pane displays the following data:

IDENTIFICADOR	IMDb
4	tt0107290
5	tt0133093
6	tt0468569
7	tt1375666
8	tt0120338
9	tt0110912
10	tt0120737
11	tt0167260
12	tt0118799
13	tt0114369
14	tt0172495
15	tt6751668

The status bar indicates: Query executed successfully. DELL_BRIAN\BEMPSQL2019 (15... DELL_BRIAN\brian (55)) proyectoCine | 00:00:00 | 16 rows

(select a la tabla PELICULA, incluyendo un count() a la tabla PELICULA)

Query executed successfully.

id_Vista	Vista_Total	Fecha	IDENTIFICADOR
9	5500	2023-01-02	9
10	5600	2023-01-02	10
11	5700	2023-01-03	11
12	5800	2023-01-03	12
13	5900	2023-01-03	13
14	6000	2023-01-03	14
15	6100	2023-01-03	15
16	6200	2023-01-04	16
17	6300	2023-01-04	17
18	6400	2023-01-04	18
19	6500	2023-01-04	19
20	6600	2023-01-04	20

(select a la tabla VISTA, incluyendo un count() a la tabla VISTA)

Query executed successfully.

Id_Usuario (PK)	Perfil_Usuario	F_Nacimiento	Domicilio	Nombre	A_Paterno	A_Materno	Gestor_Eventos	Gestor_Cartelera	Gestor_Salas	Programa_Contenido
1	Administrador General	1980-01-01	123 Calle Principal, Ciudad	Juan	Pérez	García	1	1	1	1
2	User2	1990-02-02	456 Calle Secundaria, Ciudad	Ana	López	Martínez	0	0	1	0
3	Gestor de Salas	1985-03-03	789 Calle Tercera, Ciudad	Luis	Tomé	Hernández	0	0	1	0
4	User4	1992-04-04	159 Calle Cuarta, Ciudad	María	Ruiz	Díaz	1	1	0	1
5	Coordinador de Eventos	1983-05-05	753 Calle Quinta, Ciudad	Carlos	Navarro	Molina	1	1	0	0
6	Asistente de Cartelera	1995-06-06	357 Calle Sexta, Ciudad	Sofía	Mendez	Prieto	0	1	0	0
7	User7	1988-07-07	951 Calle Séptima, Ciudad	Roberto	Blanco	Casas	0	0	0	1
8	Supervisor de Salas	1979-08-08	123 Calle Octava, Ciudad	Elena	Santos	Gómez	0	0	1	0
9	User9	1991-09-09	654 Calle Novena, Ciudad	Tomás	Fleitas	Lozano	1	0	0	1
10	Administrador de Sistema	1986-10-10	321 Calle Décima, Ciudad	Irene	García	Fernández	0	0	0	0

(select a la tabla USUARIO, incluyendo un count() a la tabla USUARIO)

ValidaTriggers


```

-- PRUEBAS TRIGGER 1
BEGIN TRANSACTION;
-- Intenta insertar un evento que no interfiera con otros eventos programados.
INSERT INTO Content.EVENTO (IDENTIFICADOR_EVENTO, QUIEN_PROGRAMA, CONTENIDO_PROGRAMADO, FECHA_HORA_INICIO, ID_SALA, IDENTIFICADOR)
VALUES (31, 'User1', 17, '2023-01-01 10:00:00.000000', 1, 16, 1);
ROLLBACK TRANSACTION;

BEGIN TRANSACTION;
-- Intenta insertar un evento que no interfiera con otros eventos programados.
INSERT INTO Content.EVENTO (IDENTIFICADOR_EVENTO, QUIEN_PROGRAMA, CONTENIDO_PROGRAMADO, FECHA_HORA_INICIO, ID_SALA, IDENTIFICADOR)
VALUES (31, 'User3', 19, '2023-01-01 14:00:00.000000', 3, 18);
ROLLBACK TRANSACTION;

select * from Content.EVENTO
select * from Content.EXHIBE

--PRUEBAS RN2
BEGIN TRANSACTION;
-- Intenta actualizar un evento que está marcado como publicado.
UPDATE Content.EVENTO
SET FECHA_HORA_INICIO = '2024-05-01 19:00:00'
WHERE IDENTIFICADOR_EVENTO = 1; -- ('validado = 1').
ROLLBACK TRANSACTION;

BEGIN TRANSACTION;
-- Intenta eliminar un evento que está marcado como publicado.

```

(Creación de las pruebas para los triggers)

Codigo Informes

```

/*
a) Un informe del monto de entradas por día por cada contenido en cartelera.
Para ello el área contable asigna un folio consecutivo del 1 al n por cada día que está
en cartelera un contenido (fecha, monto de entradas vendida).
Este informe se puede obtener también por cine
*/
select C.*, I.Ingreso_Mensual_Mayor from Administrator.CINE as C
left join ( select IDENTIFICADOR_CINE, max(INGRESO_MENSUAL) as 'Ingreso_Mensual_Mayor' from Administrator.INGRESO
group by IDENTIFICADOR_CINE ) as I
on C.IDENTIFICADOR_CINE = I.IDENTIFICADOR_CINE

```

Informe A entradas por cine

```

-- b) Top ten de las películas más vistas en un periodo de tiempo, incluyendo la cantidad monto.
select top 10
C.TITULO as Pelicula,
sum(V.Vista_Total) as Total_Vistas
from
Content.VISTA as V
inner join
Content.EVENTO as E on V.IDENTIFICADOR_EVENTO = E.IDENTIFICADOR_EVENTO
inner join
Content.EXHIBE as X on E.ID_SALA = X.ID_SALA and E.CONTENIDO_PROGRAMADO = X.IDENTIFICADOR
inner join
Content.CONTENIDO as C on X.IDENTIFICADOR = C.IDENTIFICADOR
inner join
Content.PELICULA as P on C.IDENTIFICADOR = P.IDENTIFICADOR
where
V.Fecha between '2023-01-01' and '2023-06-02'
group by
C.TITULO
order by
Total_Vistas desc;

```

Informe B top ten películas

```
--c) Ingresos por cine (todos los cines, nombre y dirección) en una fecha o rango de fechas

select C.NOMBRE, C.DIRECCION, I.INGRESO_MENSUAL, i.MES from Administrator.CINE as C
inner join Administrator.INGRESO as I
on C.IDENTIFICADOR_CINE = I.IDENTIFICADOR_CINE
where I.MES between '2024-01-01' and '2024-03-01' -- RANGO DE FECHAS VARIABLE
```

Informe C ingresos por cine

```
--d) Listado de géneros, numero de vistas ordenados de mayor a menor

select G.TIPO_GENERO, C.Numero_Genero from Content.GENERO as G
inner join (select ID_GENERO, count(ID_GENERO) as 'Numero_Genero' from Content.CONTENIDO -- encontrar los generos dentro de los contenidos
group by ID_GENERO) as C
on G.ID_GENERO = C.ID_GENERO
order by Numero_Genero desc
```

Informe D listado de género

```
--e) Listado de los usuarios del sistema, nombre completo, fecha de nacimiento y domicilio
--π(NOMBRE + A_PATERNO + A_MATERNO, F_Nacimiento, Domicilio)(USUARIO)

select NOMBRE + ' ' + A_PATERNO + ' ' + A_MATERNO as 'Nombre Completo', F_Nacimiento, Domicilio
from UserManagment.USUARIO
```

Informe E listado de usuarios

```
/*
f) Inventario de los cines: nombre, dirección, sala, número de asientos,
tipo de sala (normal, 2d, 3d, 4d, IMAX, vip, junior)
*/
--π(CINE.NOMBRE, CINE.DIRECCION, SALA.NOMBRE, SALA.CAPACIDAD, SALA.TIPO_DE_SALA (CINE M(CINE.IDENTIFICADOR_CINE = SALA.IDENTIFICADOR_CINE) SALA)

select C.NOMBRE as 'Nombre Cine', C.DIRECCION, S.NOMBRE as 'Nombre Sala', S.CAPACIDAD as 'número de asientos', S.TIPO_DE_SALA from Administrator.CINE as C
inner join Administrator.SALA as S
on C.IDENTIFICADOR_CINE = S.IDENTIFICADOR_CINE
```

Informe F inventario de cines con salas

Código Carga Inicial

```
begin tran
INSERT INTO UserManagment.USUARIO (Id_Usuario, Perfil_Usuario, F_Nacimiento, Domicilio, Nombre, A_Paterno, A_Materno,
[Gestor_Eventos], [Gestor_Cartelera], [Gestor_Salas], [Programa_Contenido])
VALUES
('User1', 'Administrador General', '1980-01-01', '123 Calle Principal, Ciudad', 'Juan', 'Pérez', 'García', 1, 1, 1, 1),
('User2', 'Gestor de Cines', '1990-02-02', '456 Calle Secundaria, Ciudad', 'Ana', 'López', 'Martínez', 0, 0, 1, 0),
('User3', 'Gestor de Salas', '1985-03-03', '789 Calle Terciaria, Ciudad', 'Luis', 'Torres', 'Hernández', 0, 0, 1, 0),
('User4', 'Gestor de Contenidos', '1992-04-04', '159 Calle Cuarta, Ciudad', 'María', 'Ruiz', 'Díaz', 1, 1, 0, 1),
('User5', 'Coordinador de Eventos', '1983-05-05', '753 Calle Quinta, Ciudad', 'Carlos', 'Navarro', 'Molina', 1, 1, 0, 0),
('User6', 'Asistente de Cartelera', '1995-06-06', '357 Calle Sexta, Ciudad', 'Sofía', 'Mendez', 'Prieto', 0, 1, 0, 0),
('User7', 'Analista de Contenido', '1988-07-07', '951 Calle Séptima, Ciudad', 'Roberto', 'Blanco', 'Casas', 0, 0, 0, 1),
('User8', 'Supervisor de Salas', '1979-08-08', '123 Calle Octava, Ciudad', 'Elena', 'Santos', 'Gómez', 0, 0, 1, 0),
('User9', 'Director Técnico', '1991-09-09', '654 Calle Novena, Ciudad', 'Tomás', 'Reyes', 'Lozano', 1, 0, 0, 1),
('User10', 'Administrador de Sistema', '1986-10-10', '321 Calle Décima, Ciudad', 'Irene', 'García', 'Fernández', 0, 0, 0, 0);
GO

select * from UserManagment.USUARIO
commit tran
```

carga de datos en la tabla (UserManagment.USUARIO)

```
--INSERTANDO 10 CINES
begin tran
-- Insertar detalles de 10 cines en la tabla CINE y sus ingresos
INSERT INTO Administrator.CINE (IDENTIFICADOR_CINE, LOGO_CORPORATIVO, LATITUD, LONGITUD, TELEFONO, DIRECCION,
NOMBRE, CODIGO_IDENTIFICACION_FISCAL)
VALUES
(1, 0x, 19.432608, -99.133209, '5555555555', 'Calle Uno 123', 'Cine Plaza', 'CIF001'),
(2, 0x, 19.434334, -99.141266, '5555555556', 'Calle Dos 456', 'Cine Sol', 'CIF002'),
(3, 0x, 19.436778, -99.142450, '5555555557', 'Calle Tres 789', 'Cine Luna', 'CIF003'),
(4, 0x, 19.438556, -99.137845, '5555555558', 'Calle Cuatro 101', 'Cine Estrella', 'CIF004'),
(5, 0x, 19.440789, -99.138916, '5555555559', 'Calle Cinco 202', 'Cine Cometa', 'CIF005'),
(6, 0x, 19.442345, -99.139987, '5555555560', 'Calle Seis 303', 'Cine Eclipse', 'CIF006'),
(7, 0x, 19.444556, -99.140356, '5555555561', 'Calle Siete 404', 'Cine Galaxia', 'CIF007'),
(8, 0x, 19.446778, -99.141425, '5555555562', 'Calle Ocho 505', 'Cine Cosmos', 'CIF008'),
(9, 0x, 19.448999, -99.142494, '5555555563', 'Calle Nueve 606', 'Cine Orion', 'CIF009'),
(10, 0x, 19.451220, -99.143563, '5555555564', 'Calle Diez 707', 'Cine Meteor', 'CIF010');
GO
```

Carga de datos en la tabla (Administrator.CINE)

```
-- Insertar registros de ingresos para varios cines
INSERT INTO Administrator.INGRESO (ID_INGRESO, INGRESO_MENSUAL, MES, IDENTIFICADOR_CINE)
VALUES
(1, 100000, '2024-01-01', 1),
(2, 150000, '2024-01-01', 2),
(3, 120000, '2024-01-01', 3),
(4, 130000, '2024-02-01', 1),
(5, 160000, '2024-02-01', 2),
(6, 140000, '2024-02-01', 3),
(7, 110000, '2024-03-01', 1),
(8, 170000, '2024-03-01', 2),
(9, 115000, '2024-03-01', 3),
(10, 125000, '2024-03-01', 4),
(11, 135000, '2024-04-01', 1),
(12, 145000, '2024-04-01', 2),
(13, 155000, '2024-04-01', 3),
(14, 165000, '2024-04-01', 4),
(15, 175000, '2024-04-01', 5);
GO

select * from Administrator.CINE
select * from Administrator.INGRESO
```

Carga de datos en la tabla (Administrator.INGRESO)

```
BEGIN TRANSACTION;
-- Insertar salas en diferentes cines con variedad de tipos de sala
INSERT INTO Administrator.SALA (ID_SALA, TIPO SOPORTE, HORA APERTURA, CAPACIDAD, NOMBRE, TIEMPO_DESCANSO, IDENTIFICADOR_CINE, TIPO_DE_SALA)
VALUES
-- Salas para el cine con IDENTIFICADOR 1
(1, 'Digital', '10:00:00', 100, 'Sala 1', 15, 1, 'Normal'),
(2, '3D', '10:00:00', 150, 'Sala 2', 15, 1, '3D'),
-- Salas para el cine con IDENTIFICADOR 2
(3, 'Digital', '11:00:00', 200, 'Sala 1', 20, 2, '2D'),
(4, '3D', '11:00:00', 120, 'Sala 2', 20, 2, '4D'),
-- Salas para el cine con IDENTIFICADOR 3
(5, 'Digital', '12:00:00', 300, 'Sala 1', 10, 3, 'IMAX'),
(6, '3D', '12:00:00', 180, 'Sala 2', 10, 3, 'VIP'),
-- Resto
(7, 'Digital', '10:30:00', 250, 'Sala 3', 12, 1, 'Junior'),
(8, '3D', '10:30:00', 220, 'Sala 4', 12, 2, 'Normal'),
(9, 'Digital', '09:00:00', 110, 'Sala 3', 15, 3, '2D'),
(10, '3D', '09:00:00', 130, 'Sala 4', 15, 1, '3D');
COMMIT TRANSACTION;
```

Carga de datos en la tabla (Administrator.SALA)

```
--INSERTANDO LAS 15 PELICULAS Y CONTENIDO EN DIRECTO
begin tran
-- Insertar géneros comunes en la tabla GENERO
INSERT INTO Content.GENERO (ID_GENERO, TIPO_GENERO)
VALUES
(1, 'Drama'),
(2, 'Crimen'),
(3, 'Acción'),
(4, 'Ciencia Ficción'),
(5, 'Romance'),
(6, 'Aventura'),
(7, 'Terror'),
(8, 'Comedia'),
(9, 'Documental'),
(10, 'Musical'),
(11, 'Deportivo');
GO
```

Carga de datos en la tabla (Content.GENERO)

```
INSERT INTO Content.CONTENIDO
(IDENTIFICADOR, AÑO, VERSION, FORMATO, TITULO, DISTRIBUIDORA, DIRECTOR, EDAD_RECOMENDADA, PAIS_PRODUCCION, TRAILER, DESCRIPCION, WEB_OFICIAL, ES_DIRECTO, DURACION, FECHA_MISION, CARTEL_ANUNCIADOR, TIPO_CONTENIDO, ID_GENERO)
VALUES
(1, 1994, 'Digital', 'Película', 'The Shawshank Redemption', 'Columbia Pictures', 'Frank Darabont', 16, 'USA', 'Two imprisoned men bond over a number of years...', 'http://shawshank.com', 0, 142, '1994-10-14', 'Drama', 1),
(2, 1972, 'Digital', 'Película', 'The Godfather', 'Paramount Pictures', 'Francis Ford Coppola', 18, 'USA', 'The aging patriarch of an organized crime dynasty transfers control...', 'http://godfather.com', 0, 176, '1972-03-24', 'Crimen', 2),
(3, 1980, 'Digital', 'Película', 'Star Wars: Episode V - The Empire Strikes Back', '20th Century Fox', 'Irvin Kershner', 12, 'USA', 'The Rebels scatter after the Empire attacks their base...', 'http://starwars.com', 0, 124, '1980-05-21', 'Ciencia Ficción', 4),
(4, 1993, 'Digital', 'Película', 'Jurassic Park', 'Universal Pictures', 'Steven Spielberg', 13, 'USA', 'During a preview tour, a theme park suffers a major power breakdown...', 'http://jurassicpark.com', 0, 127, '1993-06-11', 'Aventura', 6),
(5, 1999, 'Digital', 'Película', 'The Matrix', 'Warner Bros', 'Lana Wachowski, Lilly Wachowski', 15, 'USA', 'A computer hacker learns from mysterious rebels about the true nature...', 'http://matrix.com', 0, 116, '1999-03-31', 'Ciencia Ficción', 4),
(6, 2008, 'Digital', 'Película', 'The Dark Knight', 'Warner Bros', 'Christopher Nolan', 15, 'USA', 'When the menace known as the Joker wreaks havoc...', 'http://thedarkknight.com', 0, 152, '2008-07-18', 'Acción', 3),
(7, 2010, 'Digital', 'Película', 'Inception', 'Warner Bros', 'Christopher Nolan', 13, 'USA', 'A thief who steals corporate secrets through the use of dream-sharing technology...', 'http://inception.com', 0, 148, '2010-07-16', 'Ciencia Ficción', 4),
(8, 1997, 'Digital', 'Película', 'Titanic', 'Paramount Pictures', 'James Cameron', 12, 'USA', 'A seventeen-year-old aristocrat falls in love with a kind but poor artist...', 'http://titanic.com', 0, 195, '1997-12-19', 'Romance', 5),
(9, 1994, 'Digital', 'Película', 'Pulp Fiction', 'Miramax Films', 'Quentin Tarantino', 18, 'USA', 'The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits...', 'http://pulpfiction.com', 0, 254, '1994-10-14', 'Crimen', 2),
(10, 2001, 'Digital', 'Película', 'The Lord of the Rings: The Fellowship of the Ring', 'New Line Cinema', 'Peter Jackson', 12, 'New Zealand', 'A young Hobbit from the Shire and eight companions set out on a journey...', 'http://lordoftherings.com', 0, 178, '2001-12-19', 'Aventura', 6),
(11, 2003, 'Digital', 'Película', 'The Lord of the Rings: The Return of the King', 'New Line Cinema', 'Peter Jackson', 12, 'New Zealand', 'Gandalf and Aragorn lead the World of Men against Sauron's army...', 'http://lordoftherings.com', 0, 203, '2003-12-17', 'Aventura', 6),
(12, 1997, 'Digital', 'Película', 'Life is Beautiful', 'Miramax Films', 'Roberto Benigni', 12, 'Italy', 'When an open-minded Jewish librarian and his son become victims of the Holocaust...', 'http://lifeisbeautiful.com', 0, 116, '1997-12-20', 'Drama', 1),
(13, 1995, 'Digital', 'Película', 'Se7en', 'New Line Cinema', 'David Fincher', 18, 'USA', 'Two detectives, a rookie and a veteran, hunt a serial killer who uses the seven deadly sins...', 'http://se7en.com', 0, 127, '1995-09-22', 'Crimen', 2),
(14, 2000, 'Digital', 'Película', 'Gladiator', 'DreamWorks', 'Ridley Scott', 15, 'USA', 'A former Roman General sets out to exact vengeance against the corrupt emperor...', 'http://gladiator.com', 0, 155, '2000-05-05', 'Acción', 3),
(15, 2019, 'Digital', 'Película', 'Parasite', 'CJ Entertainment', 'Bong Joon Ho', 15, 'South Korea', 'Greed and class discrimination threaten the newly formed symbiotic relationship...', 'http://parasite.com', 0, 112, '2019-05-30', 'Drama', 1),
(16, 2022, 'Digital', 'Directo', 'Live Concert of Goldplay', 'Live Nation', 'Director N/A', 0, 'US', 'Live performance by Goldplay streamed globally.', 'http://livegoldplay.com', 1, 100, '2022-12-01', 'Concierto', 10),
(17, 2022, 'Digital', 'Directo', 'World Chess Championship', 'Chess.com', 'Director N/A', 0, 'USA', 'The annual world chess championship featuring top players.', 'http://chess.com', 1, 180, '2022-12-05', 'Deportivo', 11),
(18, 2022, 'Digital', 'Directo', 'Live Theatre: Hamlet', 'Shakespeare Theatre', 'Director N/A', 12, 'UK', 'Live performance of Hamlet from the Royal Shakespeare Theatre.', 'http://live.com', 1, 140, '2022-12-10', 'Teatro', 10),
(19, 2022, 'Digital', 'Directo', 'NBA Finals Game 7', 'ESPN', 'Director N/A', 0, 'USA', 'The final game of the NBA season live.', 'http://nba.com', 1, 120, '2022-06-15', 'Deportivo', 11),
(20, 2022, 'Digital', 'Directo', 'Met Opera Live: La Traviata', 'Metropolitan Opera', 'Director N/A', 12, 'USA', 'Live opera performance streamed from the Met.', 'http://metopera.com', 1, 180, '2022-12-20', 'Ópera', 10);
commit tran
GO
```

Carga de datos en la tabla (Content.CONTENIDO)

```
INSERT INTO Content.PELICULA (IDENTIFICADOR, IMDB)
VALUES
(1, 'tt0111161'),
(2, 'tt0068646'),
(3, 'tt0080684'),
(4, 'tt0107290'),
(5, 'tt0133093'),
(6, 'tt0468569'),
(7, 'tt1375666'),
(8, 'tt0120338'),
(9, 'tt0110912'),
(10, 'tt0120737'),
(11, 'tt0167260'),
(12, 'tt0118799'),
(13, 'tt0114369'),
(14, 'tt0172495'),
(15, 'tt6751668');
```

Carga de datos en la tabla (Content.PELICULA)

```

INSERT INTO Content.IDIOMA (ID_IDIOMA, IDIOMA, IDENTIFICADOR)
VALUES
(1, 'Inglés', 1),
(2, 'Inglés', 2),
(3, 'Inglés', 3),
(4, 'Inglés', 4),
(5, 'Inglés', 5),
(6, 'Inglés', 6),
(7, 'Inglés', 7),
(8, 'Inglés', 8),
(9, 'Inglés', 9),
(10, 'Inglés', 10),
(11, 'Inglés', 11),
(12, 'Inglés', 12),
(13, 'Inglés', 13),
(14, 'Inglés', 14),
(15, 'Inglés', 15);
GO

```

Carga de datos en la tabla (Content.IDIOMA)

```

INSERT INTO Content.DIRECTO (IDENTIFICADOR, TIPO_DIRECTO)
VALUES
(16, 'Concierto'),
(17, 'Deporte'),
(18, 'Teatro'),
(19, 'Deporte'),
(20, 'Ópera');
GO

```

Carga de datos en la tabla (Content.DIRECTO)

```

-- Insertar registros en la tabla CULTURAL
INSERT INTO Content.CULTURAL (IDENTIFICADOR, NOMBRE_COMPOSITOR, LUGAR_EMISION)
VALUES
(16, 'Coldplay', 'London, UK'),
(18, 'William Shakespeare', 'Stratford-upon-Avon, UK'),
(20, 'Giuseppe Verdi', 'New York, USA');
GO

```

Carga de datos en la tabla (Content.CULTURAL)

```

-- Insertar registros en la tabla DEPORTIVO
INSERT INTO Content.DEPORTIVO (IDENTIFICADOR, LUGAR_ORIGEN, DEPORTE_EXHIBIDO)
VALUES
(17, 'Las Vegas, USA', 'Chess'),
(19, 'Los Angeles, USA', 'Basketball');
GO

```

Carga de datos en la tabla (Content.DEPORTIVO)

```

-- Insertar registros en la tabla INTERPRETE
INSERT INTO Content.INTERPRETE (ID_INTERPRETE, NOMBRE_INTERPRETE, IDENTIFICADOR)
VALUES
(1, 'Chris Martin', 16),
(2, 'Benedict Cumberbatch', 18),
(3, 'Diana Damrau', 20);
GO

```

Carga de datos en la tabla (Content.INTERPRETE)

```

INSERT INTO Content.EXHIBE (ID_SALA, IDENTIFICADOR)
VALUES
(1, 1), -- Sala 1 exhibe "The Shawshank Redemption"
(2, 2), -- Sala 2 exhibe "The Godfather"
(3, 3), -- Sala 3 exhibe "Star Wars: Episode V - The Empire Strikes Back"
(4, 4), -- Sala 4 exhibe "Jurassic Park"
(5, 5), -- Sala 5 exhibe "The Matrix"
(6, 6), -- Sala 6 exhibe "The Dark Knight"
(7, 7), -- Sala 7 exhibe "Inception"
(8, 8), -- Sala 8 exhibe "Titanic"
(9, 9), -- Sala 9 exhibe "Pulp Fiction"
(10, 10), -- Eventos DIRECTOS
(1, 16),
(2, 17),
(3, 18),
(4, 19),
(5, 20);
GO

```

Carga datos en la tabla(Content.EXHIBE)

```

INSERT INTO Content.EVENTO (IDENTIFICADOR, QUIEN_PROGRAMA, CONTENIDO_PROGRAMADO, FECHA_HORA_INICIO, ID_SALA, IDENTIFICADOR_EVENTO)
VALUES
(1, 'User1', 16, '2023-01-01T10:00:00', 1, 16),
(2, 'User2', 17, '2023-01-01T12:00:00', 2, 17),
(3, 'User3', 18, '2023-01-01T14:00:00', 3, 18),
(4, 'User4', 19, '2023-01-01T16:00:00', 4, 19),
(5, 'User5', 20, '2023-01-01T18:00:00', 5, 20),
(6, 'User1', 16, '2023-01-02T10:00:00', 1, 16),
(7, 'User2', 17, '2023-01-02T12:00:00', 2, 17),
(8, 'User3', 18, '2023-01-02T14:00:00', 3, 18),
(9, 'User4', 19, '2023-01-02T16:00:00', 4, 19),
(10, 'User5', 20, '2023-01-02T18:00:00', 5, 20),
-- Continuar para los próximos días con el mismo patrón
(11, 'User1', 16, '2023-01-03T10:00:00', 1, 16),
(12, 'User2', 17, '2023-01-03T12:00:00', 2, 17),
(13, 'User3', 18, '2023-01-03T14:00:00', 3, 18),
(14, 'User4', 19, '2023-01-03T16:00:00', 4, 19),
(15, 'User5', 20, '2023-01-03T18:00:00', 5, 20),
(16, 'User1', 16, '2023-01-04T10:00:00', 1, 16),
(17, 'User2', 17, '2023-01-04T12:00:00', 2, 17),
(18, 'User3', 18, '2023-01-04T14:00:00', 3, 18),
(19, 'User4', 19, '2023-01-04T16:00:00', 4, 19),
(20, 'User5', 20, '2023-01-04T18:00:00', 5, 20);
GO

```

Carga de datos en la tabla (Content.EVENTO)

```

]INSERT INTO Content.VISTA (id_Vista, Vista_Total, Fecha, IDENTIFICADOR)
VALUES
(1, 5000, '2023-01-01', 1),
(2, 4500, '2023-01-01', 2),
(3, 4700, '2023-01-01', 3),
(4, 4800, '2023-01-01', 4),
(5, 5000, '2023-01-01', 5),
(6, 5200, '2023-01-02', 6),
(7, 5300, '2023-01-02', 7),
(8, 5400, '2023-01-02', 8),
(9, 5500, '2023-01-02', 9),
(10, 5600, '2023-01-02', 10),
(11, 5700, '2023-01-03', 11),
(12, 5800, '2023-01-03', 12),
(13, 5900, '2023-01-03', 13),
(14, 6000, '2023-01-03', 14),
(15, 6100, '2023-01-03', 15),
(16, 6200, '2023-01-04', 16),
(17, 6300, '2023-01-04', 17),
(18, 6400, '2023-01-04', 18),
(19, 6500, '2023-01-04', 19),
(20, 6600, '2023-01-04', 20);
GO

```

Carga de datos en la tabla (Content.VISTA)

Tabla

NOMBRE	ARCHIVO	OBSERVACIONES
DDL	crearBase.sql	<ul style="list-style-type: none"> - 15 tablas - 3 esquemas - Uso default en tablas como Evento - Uso de PK artificiales y naturales
DCL	seguridad.sql	<ul style="list-style-type: none"> - Creación de usuarios - Asignación de permisos y roles - Asignación de contraseñas
DML	dml.sql	<ul style="list-style-type: none"> - 2 triggers - 2 funciones - 6 store procedures - 4 vistas
Inserciones	cargaInicial.sql	<ul style="list-style-type: none"> - 10 cines - 15 películas - 10 salas - 20 contenidos - 30 eventos

		<ul style="list-style-type: none"> - 10 usuarios - 30 records de vistas - etc
Consultas	informes.sql	6 consultas con: -algebra relacional -JOIN -funciones de agregación -subconsultas.
Triggers	validaTriggers.sql	-2 pruebas por trigger