|  |
| --- |
| Segundo Parcial. Agosto- Diciembre 2016 |
| Documentación Base de datos proyecto |
| Brenda Samantha Ávila de la Torre 13300022 Luis Iván Morett Arévalo 13300226 José Francisco Martínez Camacho 13300195 Carlos Alejandro Rosales Celis 13300319 |
|  |
| **Programación aplicada** |
| **28/10/2016** |

|  |
| --- |
| Profesor: Ismael López Buenrostro. |

# Descripción del problema

Se desea realizar el sistema y la interfaz para el control de un auditorio donde se presentan conciertos. Este sistema tiene dos funciones generales, el punto de venta y una parte para agregar nuevos conciertos.

El auditorio se divide en zonas y cada zona tiene asientos, el precio del boleto depende de la zona en la cual se desee el asiento.

Habrá tres tipos de cuenta, una cuenta administrador desde la cual se puede agregar, editar y eliminar conciertos. El segundo tipo de cuenta será el de vendedor, el cual podrá vender los boletos para el concierto que se desee. Mientras que el tercer tipo de cuenta sera de control de acceso al concierto.

# Lista de requerimentos

1. Ninguna entrada es gratuita.

SELECT Zona.Precio FROM Zona;

1. Los conciertos están sujetos a cambios.

UPDATE *Concierto*

SET *nombre*=*value1*, *descripcion*=*value2*, *artista*=*value3,*

*genero*=*value5*, *descripcion*=*value4*

WHERE *ID\_Concierto*=*some\_value*;

UPDATE *Agenda*

SET *fecha\_inicio*=*value1*, *fecha\_fin*=*value2*

WHERE *Id\_Concierto*=*some\_value*;

1. Cada concierto tiene un horario y fecha específicos.

SELECT fecha\_inicio, fecha\_fin FROM Agenda

WHERE Agenda.Id\_Concierto=Concierto.ID\_Concierto

1. Los ID son únicos.
2. Cada zona de la sala tiene un costo fijo.

SELECT Zona.Precio FROM Zona;

1. Cada zona tiene un cupo máximo.

SELECT Zona.Cupo FROM Zona;

1. Para acceder al concierto debe de presentarse un mínimo de 10 minutos antes de que inicie el evento.
2. Boleto no transferible.
3. Se puede regresar el boleto, pero no se devuelve el dinero

DELETE FROM BOLETO WHERE ID\_Boleto=*value1*

1. Si se cancela un evento se pueden hacer devoluciones.

DECLARE @Concierto AS nchar(20);

SET @Concierto='2';

DECLARE Eliminar\_Boleto CURSOR FOR SELECT dbo.Concierto.Nombre FROM dbo.Concierto;

OPEN Eliminar\_Boleto

FETCH Eliminar\_Boleto INTO @Concierto

WHILE (@@fetch\_status = 0)

BEGIN

if (@Concierto='2')/\*1 representaria el id del concierto\*/

BEGIN

DELETE bol

FROM dbo.Boleto bol

INNER JOIN dbo.Concierto conc

ON bol.ID\_Concierto=conc.ID\_Concierto

WHERE conc.Nombre=@Concierto;

DELETE bol

FROM dbo.Concierto bol

WHERE bol.Nombre=@Concierto;

END

FETCH Eliminar\_Boleto INTO @Concierto

END

CLOSE Eliminar\_Boleto;

DEALLOCATE Eliminar\_Boleto

1. Una persona no puede entrar a una zona que no le corresponde (Control de acceso se encarga de esto).

DECLARE @id\_zona as INT;

SELECT @id\_zona=Asiento.ID\_Zona FROM Asiento

WHERE Boleto.ID\_Asiento = Asiento.ID\_Zona;

IF @id\_zona=Valor\_de\_zona\_que\_desea\_acceder

SELECT ‘true’ AS retorno;

ELSE

SELECT ‘false’ AS retorno;

1. Los asientos están señalados por fila y el número que le corresponda en esta.

SELECT asiento.Numero, Asiento.Fila FROM Asiento;

1. Solo puede ser ocupado un asiento por persona para un concierto específico.

SELECT Boleto.ID\_Boleto, Concierto.Nombre, Zona.Ubicacion, Asiento.Fila, Asiento.Numero FROM Boleto INNER JOIN Zona, Asiento, Concierto ON Boleto.ID\_boleto=Value WHERE Asiento.ID\_zona=Zona.ID\_zona AND Concierto.ID\_concierto=Boleto.ID\_Concierto and Boleto.ID\_Asiento=Asiento.ID\_Asiento ;

1. Los conciertos pueden durar más de un día.
2. En caso de durar más de un día, esto debe de haberse establecido en la agenda.

INSERT INTO Agenda(ID\_Concierto, Fecha\_Inicio, Fecha\_Fin) VALUES (‘ID del concierto en registro’,’Fecha y hora de inico’,’Fecha y hora de fin’);

1. Se registra el horario del concierto.

INSERT INTO Agenda(ID\_Concierto, Fecha\_Inicio, Fecha\_Fin) VALUES (‘ID del concierto en registro’,’Fecha y hora de inico’,’Fecha y hora de fin’);

1. El boleto cuenta con el folio de la compra.

SELECT Boleto.Folio\_Combra From Boleto;

1. Se generan folios únicos de pago que queda registrado en el boleto

DECLARE @folio int;

SELECT TOP 1 @folio= Recibo.Folio\_Compra FROM Recibo order by Folio\_Compra DESC;

INSERT INTO Boleto (‘ID\_Concierto’,’ID\_Admin’. ‘ID\_Trabajador’, ‘ID\_asiento’,‘Folio\_Compra’)

VALUES(‘ID del concierto’,’ID del admin’,’ID del trabajador’,’ID del asiento’,@folio);

1. El concierto dispone de la información necesaria para identificarla como el género musical, el artista (o grupo), nombre.

SELECT Concierto.Nombre, Concierto.Artista, Concierto.Genero FROM Concierto;

1. Una persona puede comprar más de un boleto.
2. Existen diferentes tipos de zona, y por ende tienen distintos tipos de precio.

SELECT Zona.Precio FROM Zona;

1. Entre más cerca del escenario la zona tiene un precio más elevado.
2. Existe un límite de asientos por zona.

SELECT Zona.Cupo From Zona;

1. Si se llenan los asientos de una zona, ya no se pueden vender boletos para esa zona.

DECLARE @cupo int;

SELECT @cupo= Zona.Cupo From Zona where Zona.ID\_zona=”zona donde se buscara”;

If @cupo = 0

PRINT ‘Ya no hay asientos disponibles en esta zona :c’;

1. Un boleto solo puede ser de una persona.
2. Un concierto no puede iniciar antes de la hora establecida.
3. Un concierto no puede durar más del tiempo establecido.
4. Un concierto se puede posponer 15 minutos como máximo después de la hora de inicio registrada.
5. El usuario de tipo vendedor solo puede vender boletos (y si los están devolviendo, eliminarlos) y proporcionará el boleto a la persona una vez que haya recibido el pago.
6. Si se devuelve un boleto, el vendedor solo da de baja el boleto basándose en el folio de la compra.

DELETE FROM BOLETO WHERE Boleto.Folio\_Compra=*value1*;

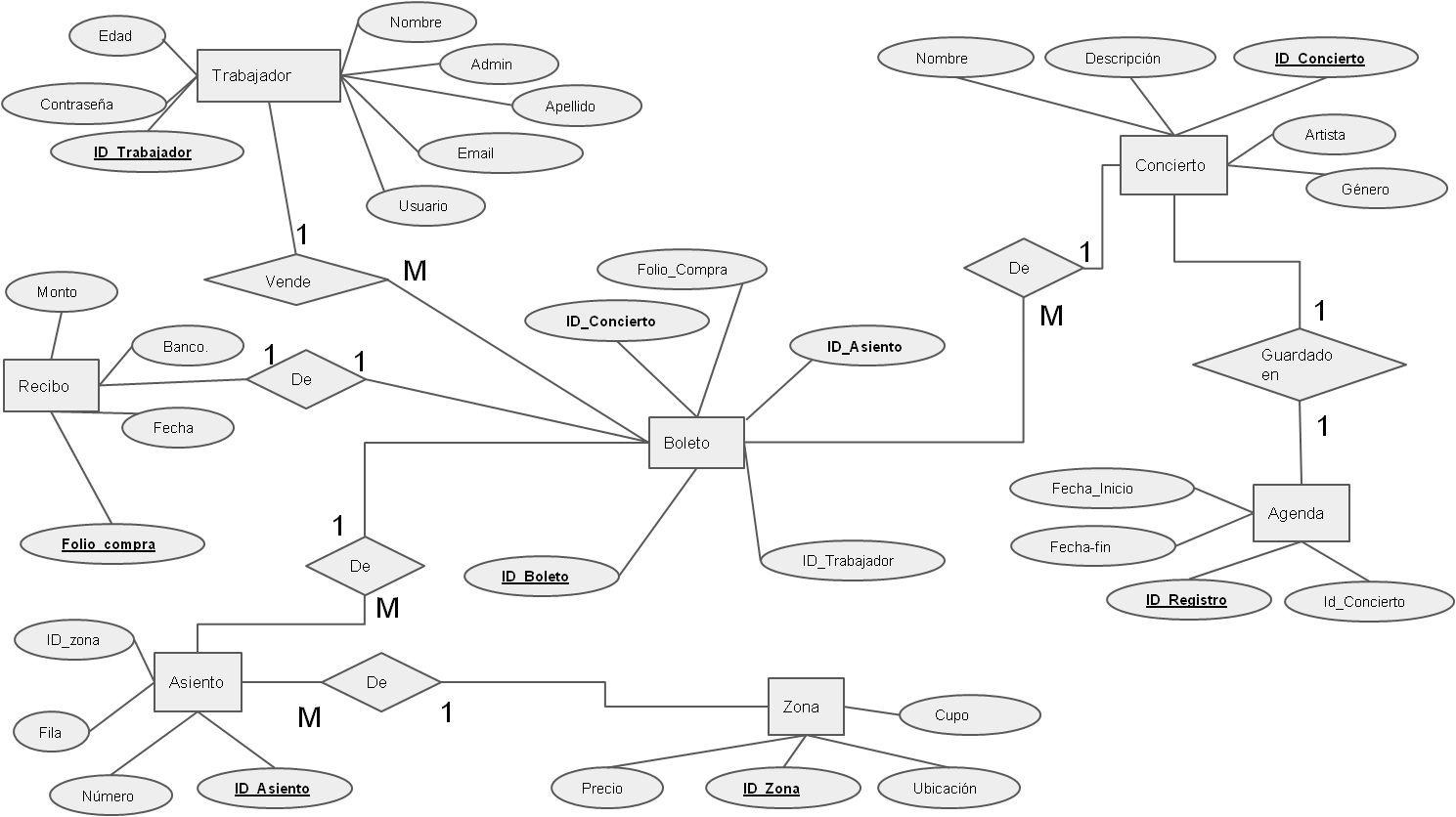
1. El usuario de tipo administrador puede agregar, modificar y eliminar conciertos del sistema según lo requiera además de en caso extraordinario realizar la venta de boletos como si fuera un usuario de tipo vendedor

INSERT INTO Concierto(Nombre,Descripcion, Artista, Genero) VALUES (’Nombre’, ’Descripcion’, ’Artista’, ’Genero’);

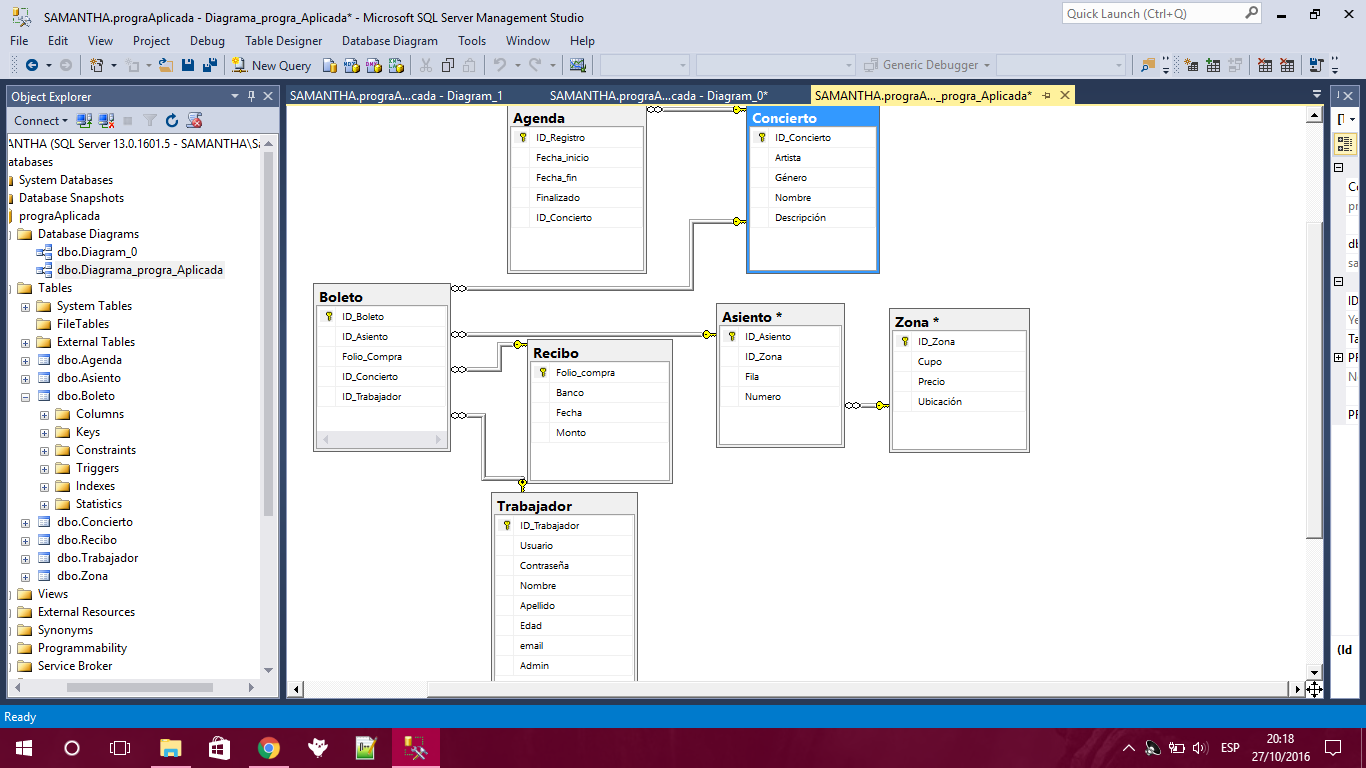
UPDATE Concierto SET Nombre=’Nombre’, Descripcion=’Descripcion’, Artista=’Artista’, Genero=’Genero’ WHERE Concierto.ID\_Concierto=value;

Para eliminar se procede de la misma manera que en el requisito 11.

# Diagrama ER



# Diagrama SQL Server



# Diagramas descriptivos

