BASES DE DATOS

PROYECTO INTEGRADOR

STEFANÍA LISSA

Índice

Enunciado y Analisis Del Problema	3
Creación Del Modelo Entidad-Relación	4
Paso Al Modelo Relacional	7
mplantación Del Modelo Físico En SQL	9
ngreso De Datos De Prueba	12
Consultas Clave	15
Vistas	15
mplementación de Procedimientos	18
Implementación de Funciones	20
Implementación de Triggers	22

Enunciado y Análisis Del Problema

Para la base de datos del programa de conciertos de música clásica del auditorio nacional, se pidió crear una estructura en la cual se pudiera almacenar tanto a los solistas como a las orquestas y de forma relacional con las obras que estas interpretaron en los conciertos a lo largo de las temporadas que tocasen en el auditorio.

Una obra debe ser interpretada por al menos un solista o una orquesta, pueden agruparse de diversas formas sin ninguna limitación. Por ejemplo, dos solistas, dos orquestas, una orquesta y varios solistas, etc. Otro requisito en cuanto a las obras es que solamente son escritas por un solo compositor.

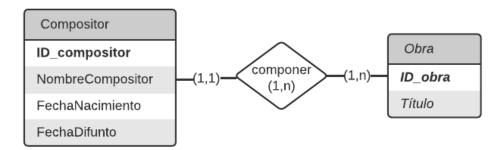
Por otro lado, una orquesta cuenta con un solo director titular, el cual liderará a la orquesta por lo menos en una de las temporadas ejecutadas en el auditorio nacional. Además, los directores pueden ser invitados por la orquesta a conciertos en los que el invitado no forme parte.

Cada persona que forme parte de una obra y los eventos que tomen lugar en el auditorio, tendrán un identificador en la base de datos y se guardarán datos característicos de su rol.

Creación Del Modelo Entidad-Relación

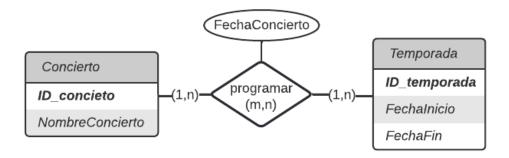
La entidad Compositor contiene además del identificador, los atributos pedidos con excepción a la cantidad de obras compuestas por el autor, porque se van a poder contar sus obras compuestas e interpretadas en el anfiteatro mediante una consulta.

Esta entidad está relacionada con la entidad Obra mediante una cardinalidad de 1 a varios porque las obras a almacenar son escritas por un solo compositor y así mismo este puede componer muchas de ellas. La entidad Obra cuenta con un identificador y un atributo para el título.

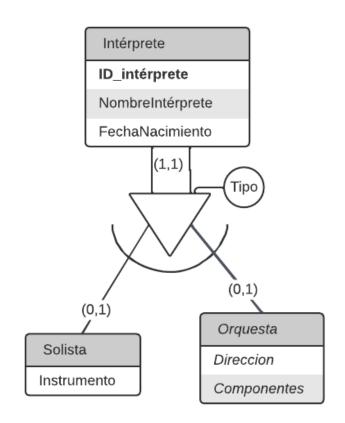


Por otra parte, he visto necesaria una entidad Temporada con la que poder mantener un mejor orden a futuro. Esta entidad tiene un identificador y dos atributos de fechas para establecer los límites de la temporada en el tiempo. En estas temporadas, toman lugar conciertos programados en fechas concretas.

Siendo así, Concierto una entidad relacionada a Temporada con una cardinalidad m,n, es decir, de muchos a muchos. Por más que no sea común, mediante esta relación se permite que los conciertos puedan ser realizados en más de una temporada. Por su parte, Concierto cuenta con un identificador y nombre.

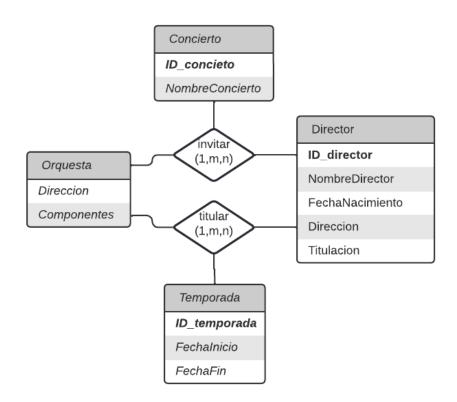


Para los solistas y orquestas he decido crear una jerarquía que permite simplificar su relación con obras y conciertos. La superclase llamada Intérprete cuenta con los atributos comunes de los solistas y orquestas. Al igual que Tipo, para diferenciar las subclases. Luego, cada uno tiene sus atributos característicos. Por un lado, de los solistas va a ser necesario saber el instrumento en el que se especializa y de la orquesta, su dirección y el número de componentes.

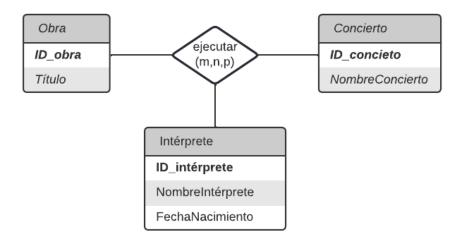


Por su cuenta, la Orquesta necesita un director que sea titular, por lo que se relaciona mediante una relación titular con una nueva entidad, Director. Este cuenta con su identificador, nombre, fecha de nacimiento, dirección y titulación. Además, mantiene su relación con la Orquesta mediante Temporada, que permite localizar a los directores en el tiempo. Por otro lado, el Director puede ser invitado por Orquesta por medio de una relación invitar a Concierto.

Ambas relaciones de Director tienen la cardinalidad 1,m,n que permite una gran cantidad de vinculaciones. pero mediante las cardinalidad mínimas se ajusta el hecho de que las orquestas solo pueden tener un titular y que un director solo puede ser invitado una vez a un concierto en específico.

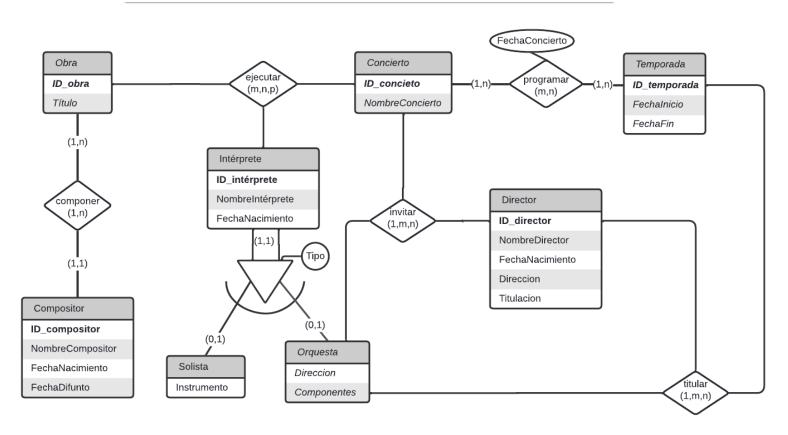


Por último en este diagrama, volvemos a la relación ternaria entre Obra, Concierto e Intérprete. Esta relación tiene una cardinalidad de muchos a muchos. Con la restricción que no se muestra mediante la cardinalidad, que es que una obra no se debe repetir en un mismo concierto.



Se muestra el diagrama entidad relación completo:

Programa de Conciertos de Música Clásica del Auditorio Nacional



Paso Al Modelo Relacional

En principio, todas las entidades van a tener su propia tabla con su identificación de clave principal, representada en negrita, con excepción de los subtipos de Solista y Orquesta. Y a continuación se explicarán las demás situaciones.

Conciertos		
ID_concierto Nombre		

Compositores					
ID_compositor Nombre FechaNacimiento FechaDifunt					

Temporadas				
ID_temporada Fechalnicio FechaFin				

Obras				
ID_obra	ID_compositor			

Debido a la relación m,n entre las entidades Concierto y Temporada, se crea una tabla aparte en la que se emplean las claves primarias de cada entidad en esta tabla como una clave primaria compuesta. Además se le agrega el campo FechaConcierto a la clave primaria para que se puedan repetir la programación de un concierto en una misma temporada. Aparte, esta tabla se llamará Itinerario para mayor comprensión.

Itinerario			
ID concierto ID temporada FechaConciert			

Para la jerarquía, se agregan los atributos que no son comunes a la tabla de la superclase con el atributo discriminante Tipo que nos ayudará a restringir los campos respectivos a si se trata de un solista o de una orquesta. Como la jerarquía es total, este atributo no puede admitir nulos y se mantiene una muy buena semántica.

Intérpretes						
ID_intérprete Tipo Nombre FechaNacimiento Instrumento Dirección Component			Componentes			

Directores				
ID_director	Nombre	FechaNacimiento	Dirección	Titulación

A causa de las relaciones terciarias, además de la tabla Directores, se generan otras dos en las que la entidad con cardinalidad 1, no forma parte de la clave primaria de la tabla.

DirectoresInvitados				
ID director ID_interprete ID concierto				

Titulares			
ID_director	ID interprete	ID temporada	

Por último, la relación ternaria entre Obra, Concierto e Intérprete, generan la siguiente tabla, que adquiere las claves primarias de las tres entidades formando una conjunta.

Ejecuciones			
ID obra	ID concierto	ID interprete	

-);

Implantación Del Modelo Físico En SQL

En estas demostraciones, el software de la base de datos va a estar construido con Oracle Database XE 11g en el sistema operativo Ubuntu 18 a partir de scripts adjuntos con la documentación.

Para comenzar se concluye el orden en el que tanto las tablas deben ser creadas, como los datos deben ser ingresados a causa de las claves foráneas. También se reflexiona sobre la cantidad de datos / espacio de almacenamiento que cada campo necesitará y que satisfacen las necesidades del auditorio.

A continuación se harán algunas observaciones sobre el script con el que se crearán las 11 tablas ya indicadas.

Como podemos ver en la imagen inferior, primordialmente en nuestra base de datos se va a poder almacenar hasta casi un millón de compositores con espacio suficiente para nombres.

```
□CREATE TABLE Compositores (
ID_compositor NUMBER(6),
Nombre VARCHAR2(50),
FechaNacimiento date,
FechaDifunto date,
CONSTRAINT PK_Compositores PRIMARY KEY (ID_Compositor),
CONSTRAINT CK_FechaDifunto CHECK(FechaDifunto>FechaNacimiento)

);
```

Luego, en cuanto a las obras, se podrían retener hasta casi cien millones de obras y con su compositor directamente vinculado sin excepciones, controlado mediante una restricción del tipo *check*.

```
□CREATE TABLE Obras(
                  NUMBER(9),
 ID obra
 Titulo
                  VARCHAR2(50),
 ID_compositor
                  NUMBER(6),
 CONSTRAINT PK Obra PRIMARY KEY (ID Obra),
 CONSTRAINT FK ID compositor Obras FOREIGN KEY (ID compositor)
 REFERENCES Compositores(ID compositor).
 CONSTRAINT CK ID compositor Obras CHECK (ID compositor IS NOT NULL)
-);
□CREATE TABLE Conciertos (
 ID_concierto
                  NUMBER(5),
 Nombre
                  VARCHAR2(50),
 CONSTRAINT PK Conciertos PRIMARY KEY (ID concierto)
```

```
□CREATE TABLE Temporadas (
 ID temporada NUMBER(4),
 Fechalnicio
                  date,
 FechaFin
                  date.
 CONSTRAINT PK Temporadas PRIMARY KEY (ID temporada),
 CONSTRAINT CK Fechas Temporadas CHECK(Fechalnicio<FechaFin)
L);
□CREATE TABLE Itinerario(
 ID concierto
                  NUMBER(5),
 ID temporada
                  NUMBER(4),
 FechaConcierto
                  date,
 CONSTRAINT PK Itinerario PRIMARY KEY (ID concierto, ID temporada, FechaConcierto),
 CONSTRAINT FK ID temporada Itinerario FOREIGN KEY (ID temporada)
 REFERENCES Temporadas(ID temporada),
 CONSTRAINT FK ID concierto Itinerario FOREIGN KEY (ID concierto)
 REFERENCES Conciertos(ID concierto).
 CONSTRAINT CK ID concierto Itinerario CHECK (ID concierto IS NOT NULL),
 CONSTRAINT CK ID temporada Itinerario CHECK (ID temporada IS NOT NULL)
L);
```

Como vemos en la imagen inferior, podremos guardar hasta casi 100 mil directorios.

```
CREATE TABLE Directores (
ID_director NUMBER(5),
Nombre VARCHAR2(50),
FechaNacimiento date,
Dirección VARCHAR2(50),
Titulación VARCHAR2(50),
CONSTRAINT PK_Directores PRIMARY KEY (ID_director)
);
```

A continuación se muestra la tabla Intérpretes con sus respectivas restricciones para solistas y orquestas, siendo estas además las únicas opciones.

```
□CREATE TABLE Intérpretes (
                 NUMBER(5),
 ID Intérprete
 Tipo
                 VARCHAR2(9),
 Nombre
                 VARCHAR2(50),
 FechaNacimiento date,
 Instrumento
              VARCHAR2(15),
                  VARCHAR2(50),
 Dirección
 Componentes
                 NUMBER(2).
 CONSTRAINT PK Intérpretes PRIMARY KEY (ID Intérprete),
 CONSTRAINT CK_Tipo CHECK (Tipo IN('Solista', 'Orquesta')),
CONSTRAINT CK Solistas Intérpretes CHECK( NOT(Tipo= 'Solista'
 AND Instrumento IS NULL AND Componentes IS NOT NULL AND Dirección IS NOT NULL)),
CONSTRAINT CK Orquesta Intérpretes CHECK( NOT (Tipo='Orquesta'
 AND Instrumento IS NOT NULL AND Componentes IS NULL AND Dirección IS NULL))
L);
```

```
CREATE TABLE DirectoresInvitados(
 ID director NUMBER(5),
 ID intérprete
                 NUMBER(5),
 ID_concierto NUMBER(5),
 CONSTRAINT PK DirectoresInvitados PRIMARY KEY (ID director, ID concierto),
 CONSTRAINT FK ID director Invitados FOREIGN KEY (ID director)
 REFERENCES Directores(ID director),
 CONSTRAINT CK ID director Invitados CHECK (ID director IS NOT NULL),
 CONSTRAINT FK ID intérprete Invitados FOREIGN KEY (ID intérprete)
 REFERENCES Intérpretes (ID intérprete),
 CONSTRAINT CK ID intérprete Invitados CHECK (ID intérprete IS NOT NULL),
 CONSTRAINT FK ID concierto Invitados FOREIGN KEY (ID concierto)
 REFERENCES Conciertos(ID concierto),
 CONSTRAINT CK ID concierto Invitados CHECK (ID concierto IS NOT NULL)
L);
□CREATE TABLE Titulares(
 ID_director
                 NUMBER(5),
 ID intérprete
                 NUMBER(5),
 ID temporada NUMBER(4),
 CONSTRAINT PK_Titulares PRIMARY KEY (ID_Intérprete, ID_temporada),
 CONSTRAINT FK ID director Titulares FOREIGN KEY (ID director)
 REFERENCES Directores(ID director),
 CONSTRAINT CK ID director Titulares CHECK (ID director IS NOT NULL),
 CONSTRAINT FK ID intérprete Titulares FOREIGN KEY (ID intérprete)
 REFERENCES Intérpretes (ID intérprete),
 CONSTRAINT CK ID intérprete Titulares CHECK (ID intérprete IS NOT NULL),
 CONSTRAINT FK ID temporada Titulares FOREIGN KEY (ID temporada)
 REFERENCES Temporadas(ID temporada),
 CONSTRAINT CK ID temporada Titulares CHECK (ID temporada IS NOT NULL)
□CREATE TABLE Ejecuciones(
 ID obra
                  NUMBER(9),
 ID Intérprete
                  NUMBER(5),
                  NUMBER(5),
 ID concierto
 CONSTRAINT PK Ejecuciones PRIMARY KEY (ID obra, ID Intérprete, ID concierto),
 CONSTRAINT FK ID obra Ejecuciones FOREIGN KEY (ID obra)
 REFERENCES Obras(ID obra),
 CONSTRAINT CK ID obra Ejecuciones CHECK (ID obra IS NOT NULL),
 CONSTRAINT FK ID concierto Ejecuciones FOREIGN KEY (ID concierto)
 REFERENCES Conciertos(ID concierto),
 CONSTRAINT CK ID concierto Ejecuciones CHECK (ID concierto IS NOT NULL).
 CONSTRAINT FK ID intérprete Ejecuciones FOREIGN KEY (ID intérprete)
 REFERENCES Intérpretes (ID intérprete),
 CONSTRAINT CK ID intérprete Ejecuciones CHECK (ID intérprete IS NOT NULL)
 -);
```

Ingreso De Datos De Prueba

Los datos a ingresar incluyen los siguientes registros y al consultarlos por comando se ven de la siguiente forma:

Cuatro compositores.

```
INSERT INTO Compositores VALUES (1, 'Compositor1', TO_DATE( '05-12-1930', 'DD-MM-YYYY'), TO_DATE( '16-11-2020', 'DD-MM-YYYY'));
INSERT INTO Compositores VALUES (2, 'Compositor2', TO_DATE( '06-11-1940', 'DD-MM-YYYY'), null);
INSERT INTO Compositores VALUES (3, 'Compositor3', TO_DATE( '07-10-1950', 'DD-MM-YYYY'), TO_DATE( '16-11-2010', 'DD-MM-YYYY'));
INSERT INTO Compositores VALUES (4, 'Compositor4', TO_DATE( '08-09-196', 'DD-MM-YYYY'), null);

SQL> SELECT * FROM COMPOSITORES;

ID_COMPOSITOR NOMBRE

FECHANAC FECHADIF

1 Compositor1

2 Compositor2

3 Compositor3

66/11/40

67/10/50 16/11/10

68/09/96
```

Nueve obras.

```
SQL> SELECT * FROM OBRAS;
```

```
INSERT INTO Obras VALUES (1, 'Obra1',1);
                                              ID_OBRA TITULO
                                                                                                  ID_COMPOSITOR
INSERT INTO Obras VALUES (2, 'Obra2',2);
                                                   1 Obra1
INSERT INTO Obras VALUES (3, 'Obra3',3);
                                                   2 Obra2
INSERT INTO Obras VALUES (4, 'Obra4',4);
                                                   3 Obra3
                                                   4 0bra4
INSERT INTO Obras VALUES (5, 'Obra5',1);
                                                   5 Obra5
INSERT INTO Obras VALUES (6, 'Obra6',1);
                                                   6 Obra6
INSERT INTO Obras VALUES (7, 'Obra7',2);
                                                   7 Obra7
                                                   8 Obra8
INSERT INTO Obras VALUES (8, 'Obra8',3);
                                                   9 Obra9
INSERT INTO Obras VALUES (9, 'Obra9',1);
```

Tres conciertos.

```
SQL> SELECT * FROM CONCIERTOS;

ID_CONCIERTO NOMBRE

INSERT INTO Conciertos VALUES (1, 'Concierto1');
INSERT INTO Conciertos VALUES (2, 'Concierto2');

2 Concierto2
```

3 Concierto3

Dos temporadas.

INSERT INTO Conciertos VALUES (3, 'Concierto3');

```
INSERT INTO Temporadas VALUES (1, TO_DATE( '01-01-2020', 'DD-MM-YYYY'), TO_DATE( '01-02-2020', 'DD-MM-YYYY'));
INSERT INTO Temporadas VALUES (2, TO_DATE( '01-06-2021', 'DD-MM-YYYY'), TO_DATE( '01-01-2022', 'DD-MM-YYYY'));

SQL> SELECT * FROM TEMPORADAS;

ID_TEMPORADA FECHAINI FECHAFIN

1 01/01/20 01/02/20
2 01/06/21 01/01/22
```

Trece registros de Itinerario referente a los conciertos tocados en una temporada.

```
ID_CONCIERTO ID_TEMPORADA FECHACON
                                                                                                 1
                                                                                                                   1 01/01/20
INSERT INTO Itinerario VALUES (1, 1, TO_DATE( '01-01-2020', 'DD-MM-YYYY'));
                                                                                                                   1 07/01/20
                                                                                                 1
INSERT INTO Itinerario VALUES (1, 1, TO_DATE( '07-01-2020', 'DD-MM-YYYYY'));
                                                                                                 1
                                                                                                                   1 15/01/20
INSERT INTO Itinerario VALUES (1, 1, TO_DATE( '15-01-2020', 'DD-MM-YYYY'));
INSERT INTO Itinerario VALUES (1, 1, TO_DATE( '22-01-2020', 'DD-MM-YYYY'));
                                                                                                 1
                                                                                                                   1 22/01/20
INSERT INTO Itinerario VALUES (1, 1, TO_DATE( '01-02-2020', 'DD-MM-YYYY'));
                                                                                                                   1 01/02/20
INSERT INTO Itinerario VALUES (2, 2, TO_DATE( '01-06-2021', 'DD-MM-YYYY'));
                                                                                                 2
                                                                                                                   2 01/06/21
INSERT INTO Itinerario VALUES (3, 2, TO_DATE( '01-07-2021', 'DD-MM-YYYY'));
                                                                                                 3
                                                                                                                   2 01/07/21
INSERT INTO Itinerario VALUES (2, 2, TO_DATE( '01-08-2021', 'DD-MM-YYYY')); INSERT INTO Itinerario VALUES (2, 2, TO_DATE( '01-09-2021', 'DD-MM-YYYY'));
                                                                                                 2
                                                                                                                   2 01/08/21
INSERT INTO Itinerario VALUES (3, 2, TO_DATE( '01-10-2021', 'DD-MM-YYYY'));
                                                                                                 2
                                                                                                                   2 01/09/21
INSERT INTO Itinerario VALUES (2, 2, TO_DATE( '01-11-2021', 'DD-MM-YYYY'));
                                                                                                 3
                                                                                                                   2 01/10/21
```

SQL> SELECT * FROM ITINERARIO;

ID_CONCIERTO	ID_TEMPORADA	FECHACON
2	2	01/12/21
3	2	01/01/22

2 01/11/21

Tres directores.

```
INSERT INTO Directores VALUES (1,'Director1', TO_DATE( '01-09-1970', 'DD-MM-YYYY'), 'Direccion1', 'Titulacion1'); INSERT INTO Directores VALUES (2,'Director2', TO_DATE( '01-09-1980', 'DD-MM-YYYY'), 'Direccion2', 'Titulacion2'); INSERT INTO Directores VALUES (3,'Director3', TO_DATE( '01-09-1990', 'DD-MM-YYYY'), 'Direccion3', 'Titulacion3');
```

SQL> SELECT * FROM DIRECTORES;

ID_DIRECTOR NOMBRE	FECHANAC DIRECCIÓN	TITULACIÓN
1 Director1	01/09/70 Direccion1	Titulacion1
2 Director2	01/09/80 Direccion2	Titulacion2
3 Director3	01/09/90 Direccion3	Titulacion3

Cuatro intérpretes, dos orquestas y dos solistas.

INSERT INTO Itinerario VALUES (2, 2, TO_DATE('01-12-2021', 'DD-MM-YYYY'));

INSERT INTO Itinerario VALUES (3, 2, TO_DATE('01-01-2022', 'DD-MM-YYYY'));

```
INSERT INTO Intérpretes VALUES (1, 'Orquesta', 'Orquesta1', TO_DATE( '01-05-2010', 'DD-MM-YYYY'), null, 'Dirección4', 50); INSERT INTO Intérpretes VALUES (2, 'Orquesta', 'Orquesta2', TO_DATE( '20-02-2000', 'DD-MM-YYYY'), null, 'Dirección5', 24); INSERT INTO Intérpretes VALUES (3, 'Solista1, 'Solista1', TO_DATE( '22-06-1963', 'DD-MM-YYYY'), 'Instrumento1', null, null); INSERT INTO Intérpretes VALUES (4, 'Solista2', TO_DATE( '25-10-1976', 'DD-MM-YYYY'), 'Instrumento2', null, null);
```

SOL> SELECT * FROM INTÉRPRETES;

ID_INTERPRETE TIPO	NOMBRE	FECHANAC	INSTRUMENTO	DIRECCION
1 Orquesta 2 Orquesta 3 Solista 4 Solista	Orquesta2 Solista1			Dirección4 Dirección5

Cuatro registros de los directores invitados por una orquesta a un concierto.

```
INSERT INTO DirectoresInvitados VALUES (1, 2, 2);
INSERT INTO DirectoresInvitados VALUES (2, 2, 1);
INSERT INTO DirectoresInvitados VALUES (2, 1, 3);
INSERT INTO DirectoresInvitados VALUES (2, 1, 3);
INSERT INTO DirectoresInvitados VALUES (3, 1, 1);

2 2 1 3 3 1 1
```

Cuatro registros de los directores que titularon a una orquesta durante una/s temporada/s.

```
INSERT INTO Titulares VALUES (1, 1, 1);
INSERT INTO Titulares VALUES (2, 1, 2);
INSERT INTO Titulares VALUES (2, 2, 1);
INSERT INTO Titulares VALUES (3, 2, 2);
INSERT INTO TITULARES;
```

Diecisiete registros de las ejecuciones, es decir, la interpretación de obras en conciertos.

```
INSERT INTO Ejecuciones VALUES (1,1,1);
                                        SQL> SELECT * FROM EJECUCIONES;
INSERT INTO Ejecuciones VALUES (2,1,1);
INSERT INTO Ejecuciones VALUES (2,3,1);
                                           ID OBRA ID INTÉRPRETE ID CONCIERTO
INSERT INTO Ejecuciones VALUES (3,3,1);
                                                             1
INSERT INTO Ejecuciones VALUES (4,3,1);
                                                                           1
INSERT INTO Ejecuciones VALUES (4,4,1);
                                                 2
                                                              3
                                                                           1
INSERT INTO Ejecuciones VALUES (5,1,1);
                                                 3
                                                              3
                                                                           1
INSERT INTO Ejecuciones VALUES (5,2,1);
                                                 4
                                                              3
                                                                           1
INSERT INTO Ejecuciones VALUES (6,2,2);
                                                 4
                                                              4
                                                                           1
INSERT INTO Ejecuciones VALUES (6,3,2);
                                                 5
                                                             1
                                                                           1
INSERT INTO Ejecuciones VALUES (6,4,2);
                                                 5
                                                             2
                                                                           1
INSERT INTO Ejecuciones VALUES (7,1,2);
                                                             2
                                                                           2
                                                 6
INSERT INTO Ejecuciones VALUES (7,2,2);
                                                               3
                                                                           2
                                                 6
INSERT INTO Ejecuciones VALUES (7,3,2);
                                                              4
INSERT INTO Ejecuciones VALUES (8,4,2);
INSERT INTO Ejecuciones VALUES (9,1,3);
                                     ID_OBRA ID_INTÉRPRETE ID_CONCIERTO
INSERT INTO Ejecuciones VALUES (9,2,3);
INSERT INTO Ejecuciones VALUES (8,4,3);
                                                             2
                                                 7
                                                                           2
                                                              3
                                                                           2
                                                 7
                                                              4
                                                                           2
                                                 8
                                                 9
                                                              1
                                                                           3
                                                 9
                                                             2
                                                                           3
                                                 8
                                                                           3
```

Consultas Clave

Vistas

1. Vista de los directores invitados por una orquesta a un concierto

```
CREATE OR REPLACE VIEW VISTA_DIRECTORES_INVITADOS AS SELECT

D.ID_DIRECTOR, D.NOMBRE AS INVITADO,
I.ID_INTÉRPRETE, I.NOMBRE AS ORQUESTA,
C.ID_CONCIERTO AS CONCIERTO
FROM DIRECTORES D, INTÉRPRETES I, CONCIERTOS C
WHERE (D.ID_DIRECTOR, I.ID_INTÉRPRETE, C.ID_CONCIERTO) IN
(SELECT* FROM DIRECTORESINVITADOS)
;

SELECT* FROM VISTA_DIRECTORES_INVITADOS;
```

ID_DIRECTOR INVITADO	ID_INTÉRPRETE ORQUESTA	CONCIERTO
3 Director3 2 Director2 2 Director2 1 Director1	1 Orquesta1 1 Orquesta1 2 Orquesta2 2 Orquesta2	1 3 1 2

2. Vista de los directores titulares de una orquesta por temporada

```
CREATE OR REPLACE VIEW VISTA_DIRECTORES_TITULARES AS
SELECT

D.ID_DIRECTOR, D.NOMBRE AS TITULAR,
I.ID_INTÉRPRETE, I.NOMBRE AS ORQUESTA,
T.ID_TEMPORADA, T.FECHAINICIO || '-' || T.FECHAFIN AS TEMPORADA
FROM DIRECTORES D, INTÉRPRETES I, TEMPORADAS T
WHERE (D.ID_DIRECTOR, I.ID_INTÉRPRETE, T.ID_TEMPORADA) IN
(SELECT * FROM TITULARES)
;

SELECT * FROM VISTA_DIRECTORES_TITULARES;
```

ID_DIRECTOR TITULAR	ID_INTÉRPRETE ORQUESTA	ID_TEMPORADA TEMPORADA
2 Director2	1 Orquesta1	2 01/06/21 - 01/01/22
1 Director1	1 Orquesta1	1 01/01/20 - 01/02/20
3 Director3	2 Orquesta2	2 01/06/21 - 01/01/22
2 Director2	2 Orquesta2	1 01/01/20 - 01/02/20

3. Vista de las obras tocadas en cierto concierto por cada intérprete

```
CREATE OR REPLACE VIEW VISTA_EJECUCIONES AS

SELECT

O.ID_OBRA, O.TITULO AS OBRA,
C.ID_CONCIERTO, C.NOMBRE AS CONCIERTO,
I.ID_INTÉRPRETE, I.NOMBRE AS INTÉRPRETE

FROM OBRAS O, CONCIERTOS C, INTÉRPRETES I

WHERE (O.ID_OBRA, C.ID_CONCIERTO, I.ID_INTÉRPRETE) IN

(SELECT * FROM EJECUCIONES)

;

SELECT * FROM VISTA_EJECUCIONES;
```

ID_OBRA	OBRA ID_CONCIER	TO CONCIERTO	ID_INTÉRPRETE INTÉRPRETE
2 2 3 4	Obra1 Obra2 Obra2 Obra3 Obra4 Obra5	1 Concierto1 1 Concierto1 3 Concierto3 3 Concierto3 3 Concierto3 1 Concierto1	1 Orquesta1 1 Orquesta1 1 Orquesta1 1 Orquesta1 1 Orquesta1
5 6 6 7	Obras Obras Obras Obras Obras Obras	2 Concierto1 2 Concierto2 3 Concierto3 1 Concierto1 2 Concierto2	1 Orquesta1 1 Orquesta1 2 Orquesta2 2 Orquesta2 2 Orquesta2 2 Orquesta2
ID_OBRA	OBRA ID_CONCIEF	TO CONCIERTO	ID_INTÉRPRETE INTÉRPRETE
9	Obra7 Obra9 Obra9	3 Concierto3 1 Concierto1 2 Concierto2	2 Orquesta2 3 Solista1 3 Solista1

4. Vista de los solistas con sus atributos

```
CREATE OR REPLACE VIEW SOLISTAS AS

SELECT ID_INTÉRPRETE, NOMBRE, FECHANACIMIENTO, INSTRUMENTO
FROM INTÉRPRETES
WHERE TIPO='Solista'
;

SELECT * FROM SOLISTAS;

ID_INTÉRPRETE NOMBRE

FECHANAC INSTRUMENTO

3 Solista1
4 Solista2
22/06/63 Instrumento1
1 Instrumento2
```

5. Vista de las orquestas con sus atributos

```
CREATE OR REPLACE VIEW ORQUESTAS AS
SELECT ID_INTÉRPRETE, NOMBRE, FECHANACIMIENTO, DIRECCIÓN,
COMPONENTES
FROM INTÉRPRETES
WHERE TIPO='Orquesta'
;
SELECT * FROM ORQUESTAS;
```

6. Vista de las obras tocadas y en cual conciertos

```
CREATE OR REPLACE VIEW VISTA_OBRAS_CONCIERTOS AS SELECT
O.ID_OBRA, O.TITULO AS OBRA,
C.ID_CONCIERTO, C.NOMBRE AS CONCIERTO
FROM OBRAS O, CONCIERTOS C
WHERE (O.ID_OBRA, C.ID_CONCIERTO) IN
(SELECT ID_OBRA, ID_CONCIERTO FROM EJECUCIONES)
;

SELECT* FROM VISTA_OBRAS_CONCIERTOS;
```

ID_OBRA O	BRA ID_CONCIERTO	CONCIERTO
1 0	bra1 1	Concierto1
2 0	bra2 1	Concierto1
3 0	bra3 1	Concierto1
4 0	bra4 1	Concierto1
5 0	bra5 1	Concierto1
6 0	bra6 2	Concierto2
7 0	bra7 2	Concierto2
8 0	bra8 3	Concierto3
8 0	bra8 2	Concierto2
9 0	bra9 3	Concierto3

7. Vista de quienes interpretaron ciertas obras

```
CREATE OR REPLACE VIEW VISTA_OBRAS_INTÉRPRETES AS SELECT
O.ID_OBRA, O.TITULO AS OBRA,
I.ID_INTÉRPRETE, I.NOMBRE AS INTÉRPRETE
FROM OBRAS O, INTÉRPRETES I
WHERE (O.ID_OBRA, I.ID_INTÉRPRETE) IN
(SELECT ID_OBRA, ID_INTÉRPRETE FROM EJECUCIONES)
;

SELECT* FROM VISTA_OBRAS_INTÉRPRETES;
```

ID_OBRA	OBRA	ID_INTÉRPRETE	INTÉRPRETE
2 2 3 4 4 5 5 6 6	Obra1 Obra2 Obra2 Obra3 Obra4 Obra4 Obra5 Obra5 Obra6 Obra6	3 1 3 4 3 2 1 4 3	Orquesta1 Solista1 Orquesta1 Solista1 Solista2 Solista1 Orquesta2 Orquesta2 Orquesta1 Solista2 Solista1 Orquesta1 Orquesta1
ID_OBRA	OBRA	ID_INTÉRPRETE	INTÉRPRETE
7 7 8 9	Obra7 Obra7 Obra7 Obra8 Obra9 Obra9	2 1 4 2	Solista1 Orquesta2 Orquesta1 Solista2 Orquesta2 Orquesta2 Orquesta1

Para ver la popularidad o obras más comunes, podemos usar esta consulta que muestra por obra, la cantidad de conciertos en la que la ejecutó, la cantidad de artistas que la interpretaron y por último el total de veces tocada tomando en cuenta que la programación de los conciertos se pueden repetir por temporadas.

Codigo a continuacion:

SQL> @SQL.sql

ID_OBRA	CONCIERTOS	INTERPRETADO	VECES_TOCADA
7	24	24	576
6	24	24	576
8	16	16	256
9	16	16	256
4	10	10	100
2	10	10	100
5	10	10	100
1	5	5	25
3	5	5	25

⁹ rows selected.

```
SELECT

E.ID_OBRA,

COUNT(E.ID_CONCIERTO) AS CONCIERTOS,

COUNT(E.ID_INTÉRPRETE) AS INTERPRETADO,

COUNT(I.ID_TEMPORADA)*COUNT(E.ID_CONCIERTO) AS VECES_TOCADA

FROM EJECUCIONES E, ITINERARIO I

WHERE I.ID_TEMPORADA IN (

SELECT I.ID_TEMPORADA

FROM ITINERARIO I

WHERE I.ID_CONCIERTO=E.ID_CONCIERTO)

GROUP BY ID_OBRA

ORDER BY VECES_TOCADA DESC;
```

Implementación de Procedimientos

1. Devuelve la cantidad de obras escritas por un compositor

```
SET SERVEROUTPUT ON;
 CREATE OR REPLACE PROCEDURE NumObras
 (V_ID IN COMPOSITORES.ID_COMPOSITOR%TYPE, NUM OUT NUMBER)
 AS
⊟BEGIN
   SELECT COUNT(*) INTO NUM FROM OBRAS
   WHERE ID COMPOSITOR=V ID;
END NumObras;
 DECLARE
 NUMOBRA NUMBER;
⊟BEGIN
 NumObras(1, NUMOBRA);
 DBMS OUTPUT.PUT LINE
 ('ESTE COMPOSITOR TIENE UN TOTAL DE 'INUMOBRAII' OBRAS COMPUESTAS');
END;
- 1
SQL> @Proc.sql;
Procedure created.
ESTE COMPOSITOR TIENE UN TOTAL DE 4 OBRAS COMPUESTAS
PL/SQL procedure successfully completed.
```

2. Devuelve el número de veces que fue ejecutada una obra en conciertos

```
SET SERVEROUTPUT ON;
 CREATE OR REPLACE PROCEDURE NumObras
 (V ID IN OBRAS.ID OBRA%TYPE, NUM OUT NUMBER)
 AS
⊟BEGIN
   SELECT COUNT(*) INTO NUM FROM EJECUCIONES
   WHERE ID OBRA=V ID;
END NumObras;
 DECLARE
 NUMOBRA NUMBER;
⊟BEGIN
 NumObras(1, NUMOBRA);
 DBMS OUTPUT.PUT LINE
 ('ESTA OBRA FUE EJECUTADA UN TOTAL DE 'INUMOBRAII' VECES');
LEND;
 1
SQL> @Proc2.sql
Procedure created.
ESTA OBRA FUE EJECUTADA UN TOTAL DE 1 VECES
PL/SQL procedure successfully completed.
```

Implementación de Funciones

1. Función que pasándole una fecha, devuelve si en esta hay un concierto o no.

```
SET SERVEROUTPUT ON;
 CREATE OR REPLACE FUNCTION HAY CONCIERTO
 (V F IN DATE)
 RETURN VARCHAR2
 AS
   RESPUESTA VARCHAR2(2);
   V ID C CONCIERTOS.ID CONCIERTO% TYPE;
BEGIN
   SELECTID CONCIERTO INTO V ID C FROM ITINERARIO
   WHERE FECHACONCIERTO=V_F;
RESPUESTA:='NO';
   ELSE
   RESPUESTA:='SI';
  END IF;
   RETURN (RESPUESTA);
 END HAY CONCIERTO;
 DECLARE
 RESPUESTA VARCHAR2(2);
 RESPUESTA:=HAY_CONCIERTO(TO_DATE('01-01-2020', 'DD-MM-YYYY'));
 DBMS OUTPUT.PUT LINE
 ('EN ESTA FECHA '||RESPUESTA||' HAY CONCIERTO');
END;
 1
Function created.
EN ESTA FECHA SI HAY CONCIERTO
PL/SQL procedure successfully completed.
SQL>
```

2. Función que devuelve la última fecha libre de una temporada, pudiendo ser utilizada para decidir cuándo programar nuevos conciertos en fechas disponibles.

```
SET SERVEROUTPUT ON;
 CREATE OR REPLACE FUNCTION FECHA_LIVRE_TEMPORADA
 (V_ID IN TEMPORADAS.ID_TEMPORADA% TYPE)
 RETURN DATE
 AS
  V FECHA DATE;
  F INI DATE;
  F FIN DATE;
  V F DATE;
∃BEGIN
  SELECT FECHAINICIO, FECHAFIN INTO F_INI, F_FIN
  FROM TEMPORADAS
  WHERE ID_TEMPORADA=V_ID;
  DBMS_OUTPUT.PUT_LINE(F_INI||' '||F_FIN);
  V F:=F FIN;
  DBMS OUTPUT.PUT LINE(V F);
  WHILE(V_F>=F_INI) LOOP
  IF (V_F>F_INI AND V_F<F_FIN) THEN
   V_FECHA:=V_F;
  END IF;
  V_F:=V_F-1;
  END LOOP;
  RETURN (V_FECHA);
END FECHA_LIVRE_TEMPORADA;
 DECLARE
 V_FECHA DATE;
BEGIN
 V_FECHA:=FECHA_LIVRE_TEMPORADA(2);
 DBMS_OUTPUT.PUT_LINE('HAY UNA FECHA LIVRE EN LA TEMPORADA EL 'IIV_FECHA);
END;
1
Function created.
01/06/21 01/01/22
01/01/22
HAY UNA FECHA LIVRE EN LA TEMPORADA EL 02/06/21
PL/SQL procedure successfully completed.
```

Implementación de Triggers

1. Controla que los conciertos están dentro de los límites de tiempo (fechas) de la temporada a la que pertenecen

```
CREATE OR REPLACE TRIGGER FECHACONCIERTO ITINERARIO
 BEFORE INSERT OR UPDATE OF ID CONCIERTO ON ITINERARIO
 FOR EACH ROW
 DECLARE
   V ID TEMPORADA TEMPORADAS.ID TEMPORADA%TYPE;
  V FECHAINICIO TEMPORADAS.FECHAINICIO%TYPE;
  V FECHAFIN TEMPORADAS.FECHAFIN% TYPE;
   V FECHACONCIERTO ITINERARIO.FECHACONCIERTO% TYPE;
   V_ID_CONCIERTO CONCIERTOS.ID_CONCIERTO%TYPE;
⊟BEGIN
   V ID CONCIERTO:=:OLD.ID CONCIERTO;
   V ID TEMPORADA:=:OLD.ID TEMPORADA;
   SELECT FECHACONCIERTO
   INTO V FECHACONCIERTO
   FROM ITINERARIO
   WHERE ID CONCIERTO=V ID CONCIERTO;
   SELECT FECHAINICIO
   INTO V FECHAINICIO
   FROM TEMPORADAS
   WHERE ID_TEMPORADA=V_ID_TEMPORADA;
   SELECT FECHAFIN
   INTO V FECHAFIN
   FROM TEMPORADAS
   WHERE ID TEMPORADA=V ID TEMPORADA;
  IF (V FECHACONCIERTO < V FECHAINICIO OR V FECHACONCIERTO > V FECHAFIN) THEN
    RAISE APPLICATION ERROR(-20000,
    'LA FECHA DEL CONCIERTO NO CORRESPONDE CON LAS FEHCAS LIMITES DE LA TEMPORADA');
  END IF:
LEND;
```

Comprobación: La temporada 2 terminó el 01-01-2022 y se intenta ingresar un concierto un mes pasado la fecha del final de la temporada

```
SQL> @Trigger1.sql

Trigger created.

SQL> INSERT INTO Itinerario VALUES (3, 2, TO_DATE( '01-02-2022', 'DD-MM-YYYY'));
INSERT INTO Itinerario VALUES (3, 2, TO_DATE( '01-02-2022', 'DD-MM-YYYY'))

*

ERROR at line 1:
ORA-01403: no data found
ORA-06512: at "ALUMNO.FECHACONCIERTO_ITINERARIO", line 11
ORA-04088: error during execution of trigger 'ALUMNO.FECHACONCIERTO ITINERARIO'
```

2. Controla que no se ingresen datos erróneos en la tabla DirectoresInvitados al verificar que el director invitado no sea titular de la orquesta que lo invita al concierto

```
CREATE OR REPLACE TRIGGER DIRECTOR_INVITADO
 BEFORE INSERT OR UPDATE ON DirectoresInvitados
 FOR EACH ROW
 DECLARE
   V_ID NUMBER(5);
   ERROR_DIRECTOR EXCEPTION;
BEGIN
   SELECT ID_DIRECTOR
   INTO V ID
   FROM TITULARES
   WHERE ID INTÉRPRETE=: NEW.ID INTÉRPRETE
   AND ID_DIRECTOR=:NEW.ID_DIRECTOR;
RAISE ERROR DIRECTOR;
   END IF:
 EXCEPTION
   WHEN ERROR_DIRECTOR THEN
   DBMS OUTPUT.PUT LINE('NO PUEDE SER INVITADO POR SU PROPIA ORQUESTA');
 END;
 INSERT INTO DirectoresInvitados VALUES (1, 1, 1);
Trigger created.
NO PUEDE SER INVITADO POR SU PROPIA ORQUESTA
1 row created.
```