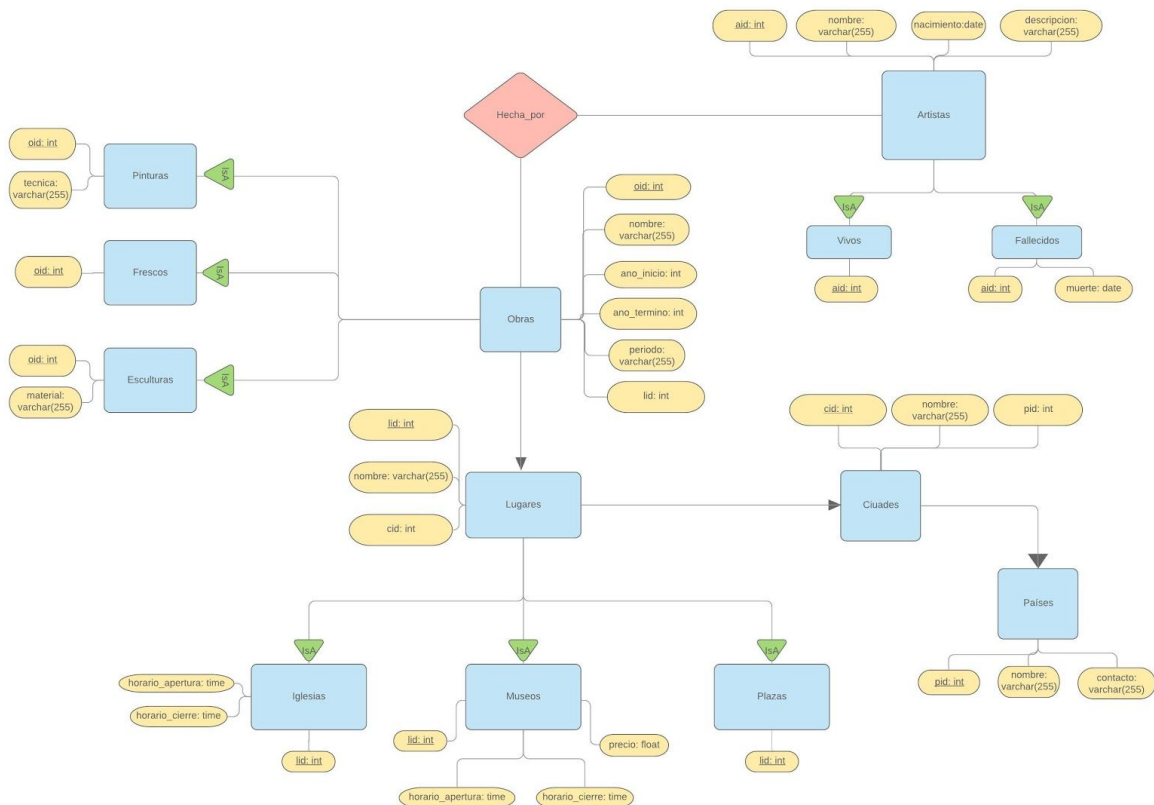


Reporte Entrega 2 Proyecto BDD

José Baboun y Matías Cadile
Grupo 12
17/04/2020

Diagrama E/R



Supuestos

El atributo “contacto” de la tabla Países se da como varchar(255) considerando que este puede venir con código de área y símbolos que no son alfanuméricos. Además, para efectos de las consultas consideramos que si el usuario no coloca nada en los cuadros de texto, se asume que quiere obtener todos los resultados posibles (a excepción de los horarios, que sí consideramos necesarios para efectos del diseño de la página).

Esquema Relacional

Artistas(aid: *int PK*, nombre: *varchar(255)* , nacimiento: *date*, descripcion: *varchar(1000)*)

Vivos(aid: *int PK FK*)

Muertos(aid: *int PK FK*, muerte: *date*)

Obras(oid: *int PK*, nombre: *varchar(255)*, ano_inicio: *int*, ano_termino: *int*, periodo:
varchar(255), lid: *int FK*)

Hecha_por(oid: *int FK*, aid: *int FK*, *PrimaryKey(oid, aid)*)

Pinturas(oid: *int PK FK*, tecnica: *varchar(255)*)

Frescos(oid: *int PK FK*)

Esculturas(oid: *int PK FK*, material: *varchar(255)*)

Lugares(lid: *int PK*, nombre: *varchar(255)*, cid: *int FK*)

Iglesias(lid: *int PK FK*, horario_apertura: *time*, horario_cierre: *time*)

Museos(lid: *int PK FK*, precio: *float*, horario_apertura: *time*, horario_cierre: *time*)

Plazas(lid: *int PK FK*)

Ciudades(cid: *int PK*, nombre: *varchar(255)*, pid: *int FK*)

Países(pid: *int PK*, nombre: *varchar(255)*, contacto: *varchar(255)*)

Modelo en 3NF

Tablas en **negrita** y dependencias funcionales enumeradas abajo. Llaves primarias subrayadas. Se dan por obvias las dependencias triviales y por tanto no se mencionan a continuación.

Obras

- oid → nombre, ano_inicio, ano_termino, periodo, lid
- nombre, ano_inicio, ano_termino, periodo, lid → oid

Pinturas

- oid → técnica

Frescos

- no hay

Esculturas

- oid → material

Artistas

- aid → nombre, nacimiento, descripción
- nombre, nacimiento, descripción → aid

Vivos

- No hay dependencias.

Fallecidos

- aid → muerte

Hecha_por

- No hay dependencias

Lugares

- lid → nombre, cid

Iglesias

- lid → horario de cierre, horario de apertura

Museos

- lid → horario de cierre, horario de apertura, precio

Plazas

- No hay dependencias

Ciudades

- cid → nombre, pid
- pid, nombre → cid

Países

- pid → nombre, contacto
- nombre, contacto → pid

- nombre → contacto

Como se puede ver, en todas las tablas de nuestro modelo tenemos relaciones de la forma $X \rightarrow Y$, donde X es la llave de la tabla o Y es parte de una llave minimal.

Tomemos como ejemplo alguna de las relaciones con dos dependencias funcionales:

Artistas. Podemos ver que aid es parte de una llave minimal (es la llave minimal), y por tanto no se rompe 3NF. Esta misma lógica se repite para las otras relaciones con dos dependencias funcionales. Además, elegimos la llave minimal como llave primaria por temas de eficiencia computacional, ya que es más óptimo cruzar información mediante el análisis de llaves numéricas que de strings (donde hay que recorrer y comparar cada carácter).

Por otro lado, tenemos un problema en la relación Países: la tercera dependencia es transitiva. En realidad, para que esté bien modelado se debería separar a la entidad Países de una entidad “Contacto_Agencia” que guarde los números de teléfono de cada país; pero como para efectos de este proyecto no nos interesa guardar la información de la agencia y la base de datos es pequeña, las ambigüedades no son relevantes, por lo que no es relevante esta dependencia y podemos decir que de todas maneras está bien implementado el 3NF.

Consultas

1. Todos los nombres distintos de las obras de arte:

```
SELECT DISTINCT Obras.nombre FROM Obras;
```

2. Todos los nombres de las plazas que contengan al menos una escultura de “Gian Lorenzo Bernini”:

```
SELECT Plazas.nombre FROM (SELECT Lugares.lid, Lugares.nombre  
FROM Lugares, Plazas WHERE Plazas.lid = Lugares.lid) as  
Plazas, (SELECT Esculturas.oid, Obras.lid FROM Esculturas,  
Obras, Hecha_por, Artistas WHERE Artistas.nombre = 'Gian  
Lorenzo Bernini' AND Esculturas.oid = Hecha_por.oid AND  
Hecha_por.aid = Artistas.aid AND Obras.oid = Esculturas.oid)  
as Foo WHERE Foo.lid = Plazas.lid;
```

3. Ingrese el nombre de un país. Muestre el nombre de todos los museos de ese país que contengan obras del renacimiento.

```
pais = input()
```

```
SELECT DISTINCT Museos.nombre FROM (SELECT Lugares.nombre,
Lugares.cid FROM Museos, Obras, Lugares WHERE Obras.periodo =
'Renacimiento' AND Obras.lid = Museos.lid AND Lugares.lid =
Museos.lid) as Museos, Ciudades, Países WHERE Museos.cid =
Ciudades.cid AND Ciudades.pid = Países.pid AND
LOWER(Países.nombre) LIKE LOWER('%país%') ;
```

4. Para cada artista, entregue su nombre y el número de obras que ha participado

```
SELECT Artistas.nombre, COUNT(Artistas.nombre) AS
numero_obras FROM Obras, Hecha_por, Artistas WHERE Obras.oid
= Hecha_por.oid AND Hecha_por.aid = Artistas.aid GROUP BY
Artistas.nombre;
```

5. Ingrese una hora de apertura, una hora de cierre (formato hh:mm:ss) y una ciudad. Muestre los nombres de las iglesias ubicadas en esa ciudad, abiertas entre esas horas (inclusive) junto a todos los nombres de los frescos que encuentra en cada una de ellas. Una fila por tupla.

```
apertura = input()
cierre = input()
ciudad = input()
```

```
SELECT Iglesias.nombre, Frescos.nombre FROM (SELECT
Lugares.lid, Lugares.nombre FROM Iglesias, Lugares, Ciudades
WHERE Iglesias.lid = Lugares.lid AND Lugares.cid =
Ciudades.cid AND Iglesias.horario_apertura <= 'apertura' AND
Iglesias.horario_cierre >= 'cierre' AND
LOWER(Ciudades.nombre) LIKE LOWER('%ciudad%')) AS Iglesias,
(SELECT * FROM Frescos, Obras WHERE Frescos.oid = Obras.oid)
AS Frescos WHERE Frescos.lid = Iglesias.lid;
```

6. Encuentre el nombre de cada museo, plaza o iglesia que tenga obras de todos los períodos del arte que existan en la base de datos.

```
SELECT Lugares.nombre FROM Lugares, Obras, (SELECT * FROM
Obras) AS Foo WHERE Lugares.lid = Obras.lid GROUP BY
Lugares.nombre HAVING COUNT (DISTINCT Obras.periodo) >= COUNT
(DISTINCT Foo.periodo);
```