

Tarea 3 - Diseño Relacional

Fundamentos de Bases de Datos - Laboratorio 2024

Objetivos:

- Trabajar los temas de diseño relacional y normalización sobre un ejemplo concreto, experimentando con las dificultades de procesar datos con un diseño de baja calidad.
- Mejorar la comprensión de la teoría del diseño relacional clásico, estudiando algunos temas puntuales como las diferencias entre el modelo real (SQL) y el modelo teórico y el tratamiento de valores nulos.

Documentación:

- Todos los documentos (libro y slides) usados en el curso.
- Manual de postgresql, versión 12.x o superior.

Entrega:

- Informe en un archivo PDF con cada una de las partes requeridas incluyendo las consultas SQL realizadas cuando corresponda.
- **IMPORTANTE.** En el informe, indicar el número de grupo y los integrantes
- Se podrán realizar entregas hasta el lunes 04/11/2024 a las 23:59.

Monitoreos:

Esta tarea tendrá dos monitoreos: la semana del 21/10 y la del 28/10.

Descripción de la Realidad: Países del Mundo y sus Líderes

DBpedia¹ es un proyecto donde se extraen y procesan datos de la Wikipedia.

En esta tarea se trabajará con datos tomados de DBpedia sobre países, líderes de los países, ciudades de los países y personas nacidas en esas ciudades.

La base de datos consta de una sola tabla denominada `países_lideres` que tiene 20 Atributos. Todas las descripciones de los atributos refieren a un tupla de la tabla, es decir, cuando dice *Identificación de una Ciudad del país* se refiere a la identificación de una ciudad del país cuya identificación aparece en la misma tupla, en el atributo `id_país`. Los atributos son los siguientes:

Atributo	Descripción
<code>id_país</code>	Identificación del país.
<code>n_país</code>	Nombre del país.
<code>d_país</code>	Descripción del país.
<code>id_capital</code>	Identificación de la capital.
<code>n_capital</code>	Nombre de la capital.
<code>d_capital</code>	Descripción de la capital.
<code>cargo</code>	Denominación del cargo del líder del país.
<code>id_lider</code>	Identificación del líder del país.
<code>n_lider</code>	Nombre del líder.
<code>bio_lider</code>	Biografía del líder.
<code>id_ciudad</code>	Identificación de una Ciudad del país.
<code>n_ciudad</code>	Nombre de la Ciudad.
<code>d_ciudad</code>	Descripción de la Ciudad.
<code>id_pers</code>	Identificación de una persona nacida en la ciudad.
<code>n_pers</code>	Nombre de una persona nacida en la ciudad.
<code>fecha_nac</code>	Fecha de nacimiento de la persona de <code>id_pers</code> .
<code>fecha_fallec</code>	Fecha de fallecimiento de la persona de <code>id_pers</code> .
<code>id_lugarfallec</code>	Identificación del lugar de fallecimiento de la persona de <code>id_pers</code> .
<code>tipo_lugarfallec</code>	Tipo del lugar de fallecimiento de la persona de <code>id_pers</code> . (país o ciudad)
<code>n_lugarfallec</code>	Nombre del lugar de fallecimiento de la persona de <code>id_pers</code> .

Se conoce además que:

- Un país puede tener más de un líder y un líder puede ser líder de más de un país.
- Un líder puede tener más de un cargo en un mismo país.
- El atributo `id_LugarFNac` puede tener una ciudad o un país. Ese tipo está indicado en el atributo `tipo_LugarFNac`. Una misma persona puede aparecer con los dos tipos de lugares de fallecimiento.

En el **repositorio** se encuentran disponibles *scripts* para crear la tabla (*esquema.sql*) así como cargar un primer conjunto de datos (*datos_1.sql*).

Para la creación de la base y la carga de los datos se recomienda seguir los siguientes pasos en la línea de comandos:

1. Crear la base de datos (`createdb -U postgres_username -E UTF8 tarea3`).
 2. Ingresar a la base creada en el paso anterior (`psql -U postgres_username -d tarea3`).
 3. Setear el encoding a utf8 (`set client_encoding to utf8;`).
Para corroborar que el encoding es correcto se puede correr el comando `show client_encoding;`.
- Observación:** Ambos comandos llevan el carácter “;” al final.

¹Proyecto DBpedia <http://dbpedia.org>

4. Ejecutar los 2 scripts de carga (*\i path/to/script.sql*), en el siguiente orden:

- a) *esquema.sql*
- b) *datos_1.sql*

Parte 1 - dependencias funcionales y claves

1. En este ítem, definiremos consultas SQL para determinar el cumplimiento de dependencias funcionales en una tabla creada usando SQL.
 - (a) Escribir 2 (dos) consultas SQL distintas (una de ellas usando la clausula *group by* y la otra no) para determinar si se cumple una dependencia funcional $X \rightarrow Y$ sobre una tabla $R(X, Y, Z)$ (sin ninguna restricción definida). Si se cumple devolver el conjunto vacío, en el otro caso devolver el conjunto de los valores de la parte izquierda, X . Explicarlas.
 - (b) Aplicar las 2 consultas anteriores para determinar si las siguientes DF se cumplen en la instancia dada de la relación *países_lideres*. Compruebe y explique el resultado obtenido.
 - I. $id_lugar_fallec \rightarrow tipo_lugar_fallec, n_lugar_fallec$
 - II. $id_pers, tipo_lugar_fallec \rightarrow id_lugar_fallec$
 - III. $id_pers \rightarrow n_pers, fecha_nac, fecha_fallec$
 - (c) Ejecute el script *datos_2.sql* que agrega nuevas tuplas. Repita el ítem anterior (b) sólo para la DF II. Compruebe y explique el resultado obtenido.
 - (d) Modificar las 2 consultas para devolver no sólo los atributos de la parte izquierda de la DF sino también los atributos de la parte derecha. Repita el ítem anterior y explique el resultado obtenido.
2. Examinando la semántica de los atributos de la tabla, las dependencias funcionales que se cumplen en esta realidad son las siguientes:
 - $id_pais \rightarrow n_pais, d_pais, id_capital$
 - $id_capital \rightarrow n_capital, d_capital, id_pais$
 - $id_ciudad \rightarrow n_ciudad, d_ciudad, id_pais$
 - $id_lider \rightarrow n_lider, bio_lider$
 - $id_pers \rightarrow n_pers, fecha_nac, fecha_fallec, id_ciudad$
 - $id_lugar_fallec \rightarrow n_lugar_fallec, tipo_lugar_fallec$
 - $id_pers, tipo_lugar_fallec \rightarrow id_lugar_fallec$

Determinar **todas las claves** de la tabla *países_lideres*. Justifique.

3. Escribir 2 (dos) consultas SQL que permitan encontrar tuplas de la instancia que violan alguna de las claves candidatas halladas en el ítem 2. Asegúrese que las tuplas violan la clave por motivos diferentes, explicando los motivos en cada caso. Considere la noción de **Clave Primaria en SQL**.
4. Observe que la tabla *países_lideres* no tiene una clave primaria declarada, intentaremos resolver esto.
 - (a) Declare una nueva tabla, llamada *pa_li2*, con los mismos atributos de la tabla *países_lideres*. SUGERENCIA: busque el uso de *like* en creación de tablas.
 - (b) Incluya la declaración de una de las claves candidatas halladas en el ítem 2 como clave primaria de la tabla *pa_li2*.
 - (c) Escriba una consulta SQL para cargar la tabla *pa_li2* desde la tabla *países_lideres* asegurándose de no perder a ningún país. Su solución no puede usar *triggers*. SUGERENCIA: busque el uso de *on conflict*. Descarte las tuplas si hay claves duplicadas y reemplace los nulos con valores que considere convenientes. Explique los criterios utilizados y sus consecuencias.

5. Uso de triggers

- (a) Declare una nueva tabla, llamada `pa_li3`, idéntica a la tabla `pa_li2`.
- (b) En este ítem, se pide cargar la tabla `pa_li3` desde la tabla `países_lideres`. Las tuplas que no puedan ser cargadas por no cumplir la definición de clave primaria, deberán ser almacenadas en una tabla `aux`, idéntica a la tabla `países_lideres`. Esta vez, escribiendo consultas SQL y *triggers*.

Parte 2 - normalización

- 8. (a) Obtener una descomposición de la tabla `países_lideres` en 3NF con join sin pérdida y preservación de dependencias aplicando el algoritmo visto en el curso.
 - (b) Revise la descomposición encontrada y sugiera otra descomposición con menor cantidad de tablas y que esté también en 3NF. Justifique.
9. Indicar al menos una dependencia multivaluada, no trivial, no funcional en alguna de las tablas resultantes de la descomposición anterior. Justifique su respuesta.
10. Indicar si la descomposición del ítem 8 viola 4NF. Si es así, aplicar el algoritmo visto en el curso para obtener una descomposición en 4NF a partir de ella. Indique si se pierden dependencias de las indicadas en la parte 1.