Diseño relacional y normalización

Alex Di Genova

18/04/2022

Outline

- Problema GoogleColab
- Repaso
- Normalización

Problema GoogleColab

Solución

In [1]:
```
!pip install SQLAlchemy==1.4.46
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting SQLAlchemy==1.4.46
  Downloading SQLAlchemy-1.4.46-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 13.6 MB/s eta 0:00:00
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.9/dist-packages (from SQLAlchemy==1.4.46) (2.0.2)
Installing collected packages: SQLAlchemy
  Attempting uninstall: SQLAlchemy
    Found existing installation: SQLAlchemy 2.0.9
    Uninstalling SQLAlchemy-2.0.9:
      Successfully uninstalled SQLAlchemy-2.0.9
Successfully installed SQLAlchemy-1.4.46
```

In [2]:
```
%load_ext sql
%sql sqlite:///ONG.db
```

In [3]:
```
%%sql
PRAGMA foreign_keys;
PRAGMA foreign_keys = ON;
PRAGMA foreign_keys;
```

```
 * sqlite:///ONG.db
Done.
Done.
Done.
```

- https://github.com/adigenova/uohdb/blob/main/code/Fix_Bug__SQLAlchemy.ipynb

Directrices de diseños informales para los esquemas de relación

Resumen de directrices

- Diseñar un esquema de relación que sea fácil explicar su significado.

- Evitar problemas de inserción, borrado y actualización en relaciones.

- Evite al máximo situar en las relaciones atributos que sean NULL frecuentemente.

- Diseñar relaciónes puedan concatenarse con condiciones de igualdad en los atributos que son parejas de clave primaria y foranea de forma que se garantice la no generación de tuplas falsas.

Dependencias funcionales

Reglas de inferencia

- Los axiomas de Armstrong son un conjunto de reglas de utilizadas para inferir todas las dependencias funcionales en una base de datos relacional. Fueron creadas por William W. Armstrong (matematico, 1974).
- **Dependencia trivial (RI1)**
    - Si Y es subconjunto de X entonces X->Y
- **Aumentación (RI2)**
    - Si X -> Y => (X U Z) -> (Y U Z) => XZ -> YZ
        - Cada atributo no clave debe depender completamente del PK.
        - Ejemplo: id_estudiante, curse -> Nombre_e,Ciudad, Provincia, Nota, fecha_completado
            - Tabla1 : id_estudiante, curso, nota, fecha_realizado
            - Tabla 2: id_estudiante, nombre_e, ciudad, provincia.
- **Transitividad (RI3)**
    - Si X -> Y and Y -> Z => X -> Z
        - Ejemplo id_estudiante -> Nombre_e, Ciudad, Provincia, Nombre_programa, id_programa
        - Tabla 1: id_estudiante -> Nombre_e, Ciudad, Provincia, id_programa
        - Tabla 2: id_programa -> Nombre_programa

Normalización

Normalización

Introducción

- Un esquema de relación por una serie de comprobaciones para certificar que satisface una determinada forma normal (Codd 1972).
- Codd propuso tres formas normales: 1FN, 2FN y 3FN
    - Basadas en las DF entre los atributos de una relación
- Posteriormente se propusieron 4FN y 5FN
    - Basadas en dependencias multivalor y concatenación.
- El proceso de normalización es un análisis de un esquema de relación, basado en sus **DF** y sus claves principales, para **garantizar** propiedades deseables como (1) **minimizar la redundancia** y (2) **minimizar las anomalías** de inserción, borrado y actualización.
- Esquemas que no satisfacen las reglas de normalizacion, se descomponen que cumplen las reglas y por consiguiente cuentan con las propiedades deseables.
- La normalización ofrece:
    - Un marco formal para analizar esquemas de relacion basado en sus claves y en las DF de sus atributos.
    - Una seria de pruebas de forma normal que pueden efectuarse sobre modelos relacionales, con el obetivo de normalizar la base de datos hasta el grado deseado (1FN,2FN,3FN).

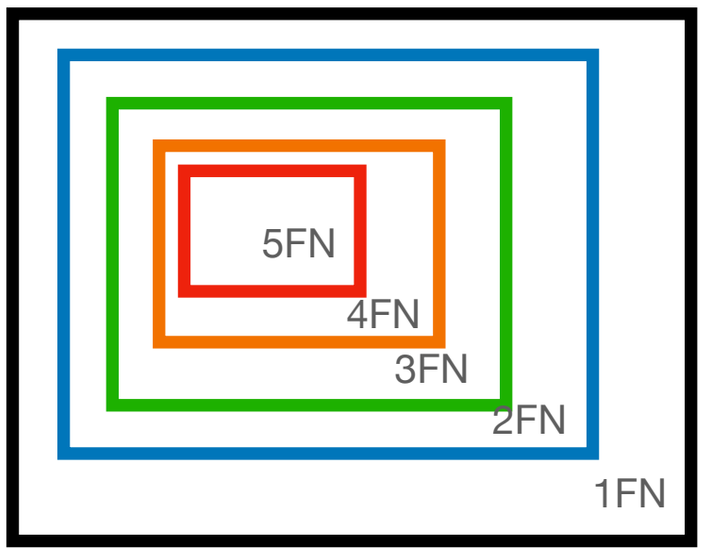
Normalización

Propiedades

- El proceso de normalización por descomposición debe preservar la existencia de las propiedades de los esquemas relacionales.

  - Propiedad de reunión no aditiva, que garantiza que no se presentará el **problema de tuplas falsas**, luego de realizar la normalización.

  - Propiedad de conservación de las dependencias, que asegura que todas las **DF están representadas** en alguna relación luego de la normalización.



Incremental

- En la práctica normalizamos hasta la 3FN (2FN por rendimiento)

Normalización

definición

- Primera forma normal (1FN)
  - Creada para prohibir atributos multivalor, compuestos y sus combinaciones.
- *Un atributo solo debe incluir valores atomicos (simples, indivisibles) y el valor de cualquier atributo en una tupla debe ser un valor simple del dominio de ese atributo.*

- Los únicos valores de atributo permitidos por la 1FN son los **atómicos** (o indivisibles)

DEPARTAMENTO

NombreDpto	NúmeroDpto	DniDirector	UbicacionesDpto
Investigación	5	333445555	{Valencia, Sevilla, Madrid}
Administración	4	987654321	{Gijón}
Sede central	1	888665555	{Madrid}

- Soluciones:
  - Eliminar el atributo que viola la 1FN y colocarlo en una relación aparte.
  - Expandir la clave primaria ({NúmeroDepto,UbicacionesDpto}, *redundancia*)
  - Si se conoce el número máximo de valores para el atributo (N), sustituir el atributo por N atributos atomicos (valores NULL, consultas más complejas).

Normalización

Definición

- Una relación se encuentra en 2FN si:
  - Cumple la 1FN.
  - *Cada atributo no clave tiene dependencia funcional completa respecto de alguna de las claves.*
- Ejemplo:
  - R(A,B,C,D) DF{A,B->C; A->D}
  - Clave candidata es {A,B} => atributos no clave C y D.
  - El atributo D depende de A pero no de una clave => el esquema no cumple 2FN.
  - Solución: R1(A,B,C) {A,B->C} y R2(A,D){A->D} si cumplen 2FN.

Normalización

Definición

- Una relación se encuentra en 3FN si:
  - Cumple la 2FN.
  - *No existe ningún atributo no principal que dependa transitivamente de alguna clave de R*

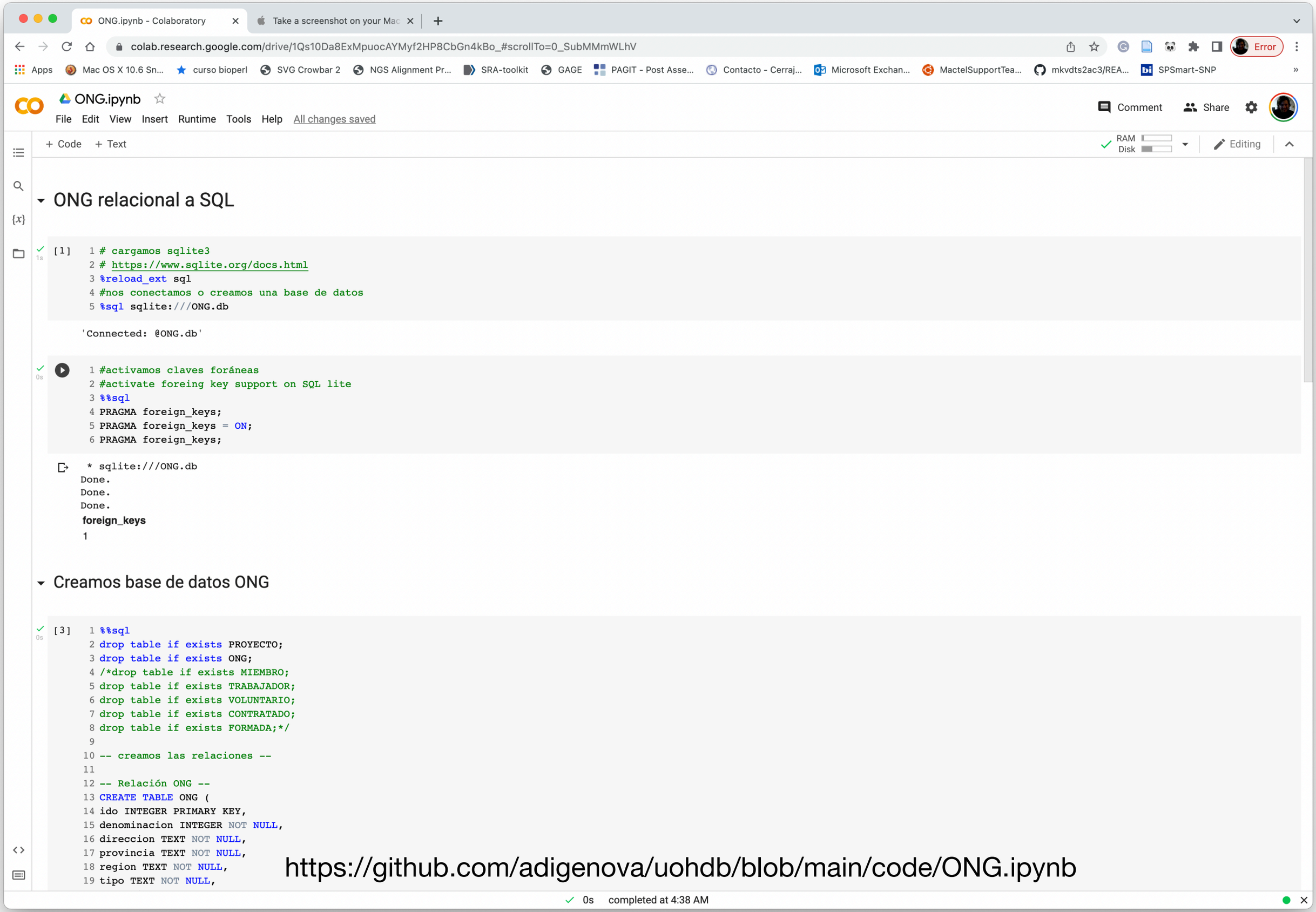
- Ejemplo:
  - R(A,B,C) DF{A->B; B->C}
  - Clave candidata es {A} => atributos no clave B y C.
  - El atributo C depende transitivamente de la clave A => el esquema no cumple 3FN
    - Si cumple 2FN, ya que C depende transitivamente de la clave (A).
  - Solución: R1(A,B) {A->B} y R2(B,C) {B->C} si cumplen 3FN.

Normalización

Resumen

FN	Prueba	Solución
1FN	La relación no debe tener atributos multivalor o relaciones anidadas.	Generar nuevas relaciones para cada atributo multivalor o relación anidada.
2FN	Cada atributo no clave tiene dependencia funcional completa respecto de alguna de las claves.	Descomponer y configurar una nueva relación por cada clave parcial con su(s) atributo(s) dependiente(s).
3FN	No existe ningún atributo no principal que dependa transitivamente de alguna clave de R	Descomponer y configurar una relación que incluya el(los) atributo(s) no clave que determine(n) funcionalmente otro(s) atributo(s) no clave.

GoogleColab



Consultas?

Consultas o comentarios?

Muchas gracias