



**Facultad de
Ciencias**
UNAM

**Fundamentos de Bases de Datos
Grupo 7077**

PRÁCTICA 8. POBLACIÓN DE BASE DE DATOS

EQUIPO: mmm xd

317088296 - Demian Oswaldo Garcia Toxqui.

317042522 - Erick Bernal Márquez.

317180321 - Karen Cristóbal Morales.

317061521 - Rosa María Robles Huerta.

317205776 - Rubén Acosta Arzate.

12 de Mayo de 2022



1. Correcciones a nuestro archivo DDL.sql

- Cambiamos la referencia de la llave foránea en **Recibo**, porque la llave primaria de "Caja" es compuesta y nosotros hacíamos la referencia a cada componente por separado cuando debe de ser a los dos como sigue:
... **FOREIGN KEY**(idCaja, idClinica)...
- Mismo caso que el anterior.
En **ClienteNormal** y **ClienteFrecuente** había una llave foránea por parte de "Caja", entonces cambiamos la referencia de las llaves foráneas de la siguiente forma:
... **FOREIGN KEY**(idCaja, idClinica)...
- Mismo caso para **GenerarReciboNormal**, **GenerarReciboEmergencia** y **GenerarReciboTratamiento**.
La llave primaria de las consultas es compuesta y habíamos hecho referencia por separado, por lo tanto la referencia de las llaves foráneas quedan de la siguiente forma:
... **FOREIGN KEY**(idConsulta, idMascota)...
- Análogo a los casos anteriores, en **VenderJuguete**, **VenderAccesorio**, **VenderMedicamento** y **VenderComida** hay una llave foránea "Caja", por lo tanto la referencia de las llaves foráneas quedan de la siguiente forma:
... **FOREIGN KEY**(idCaja, idClinica)...

2. Población de la Base de Datos

La herramienta que utilizamos para generar los datos fue mockaroo cuyo link es:

<https://mockaroo.com/>

Describiremos cómo solucionamos los casos especiales tales como curp y rfc los cuales están compuestos por el nombre de la persona así como los id o llaves foráneas, para los demás atributos los daremos por hecho ya que la herramienta permite generarlos de manera automática sin ningún problema.

La manera en que utilizamos la herramienta para cada una de las tablas fue la siguiente:

1. Supervisor

Creamos un esquema auxiliar con sus respectivos campos de la tabla solo que al último el curp y el rfc, esto para poder crear de manera eficiente los curps y rfc utilizando expresiones regulares, ya que si ponemos primero los curps, como estos están compuestos del nombre completo y fecha de nacimiento la herramienta no permite leerlo debido a que éstos campos no se definieron antes, mismo caso para el rfc que tiene que obtener los datos de campos anteriores. Este mismo principio se sigue para la generación de otros campos que requieran un formato específico. Descargamos y subimos su data set.

Luego creamos un esquema Supervisor para poner en orden los campos jalándolos del dataset auxiliar anterior.

Por último generamos los datos a partir del esquema anterior y lo subimos como dataset para tener de donde jalar los datos al crear tablas que requieran de llaves foráneas y estas en efecto existan manteniendo la integridad referencial.

Este mismo proceso de crear tablas auxiliares para la correcta generación de datos específicos, crear su data set, crear otra tabla para ponerlos en orden, generar los datos y subirlos como dataset se sigue en otras tablas que ya mencionaremos más adelante.

2. **Clinica**

Creamos un esquema con los respectivos campos definidos por la tabla, en el caso de la generación de id's utilizamos expresiones regulares donde los 3 primeros caracteres son letras al azar y los otros 2 número al azar.

Del mismo modo descargamos los datos y lo subimos como dataset para tener de donde jalar los datos en aquellas tablas que requieran del id como llave foránea.

3. **Juguete**

Siguiendo el principio de *Supervisor*

Para juguete creamos un esquema auxiliar para la correcta generación de los id's, y los nombres. Utilizamos un campo auxiliar que genere algún nombre científico y en el id ponemos los 3 primeros caracteres seguido de numeros al azar. En el caso de los juguetes tenemos las opciones de: Pelota, Mordedera, Disco, Hueso y Peluche. Descargamos su dataset y lo subimos.

Creamos un esquema con los respectivos campos de la tabla para la generación de datos donde solo jalamos el id de la tabla auxiliar anterior y de la misma manera jalamos los juguetes de la tabla auxiliar anterior.

Generamos los datos y los subimos como dataset.

4. **Accesorio**

Lo mismo que *Juguete* con los respectivos campos de juguete solo que las opciones de accesorio son: Correa, Cama, Plato, Collar, Caja.

De la misma manera descargamos los datos y los subimos como dataSet.

5. **Medicamento**

Lo mismo que en la tabla anterior, las opciones son: Vacunas, Antibioticos, Antiparasitos.

Descargamos datos y los subimos como dataset.

6. **Comida**

Mismo que el anterior, las opciones son: Alimento seco, Alimento húmedo.

Descargamos y subimos como dataset

7. **Consultorio**

Creamos un esquema con los respectivos campos de la tabla, en el caso del id lo jalamos del dataset de clinica, ya que de este aseguramos que las llaves existan manteniendo así la integridad referencial. Además el id del cuarto son números secuenciales.

Descargamos y subimos como dataset.

8. **CuartoCuidados**

Lo mismo que el caso anterior con sus respectivos campos de la tabla CuartoCuidados.

9. **Caja**

Mismo caso que *Consultorio*.

10. **TenerJuguete**

Como ambos son llaves foráneas basta con crear el esquema con sus respectivos campos y jalar el idClinica del dataset de clinica y el idProducto del dataset de juguete.

11. **TenerAccesorio**

Es el mismo caso de *TenerJuguete*, basta con crear el esquema y jalar el idClinia del dataset de Clinica y el idProducto del dataset de Accesorio.

12. **TenerMedicamento**

Mismo caso que *Tener accesorio*, así que el esquema es análogo.

13. **TenerComida**
Mismo caso, el esquema es análogo.
14. **VenderJuguete**
Como todas son llaves foráneas, basta con hacer el esquema con sus respectivos campos y jalar el idProducto del dataset de juguete, jalar el idCaja del dataset de caja y idClinica del dataset de caja.
15. **VenderAccesorio**
De la misma manera, basta con hacer el esquema con sus respectivos campos y jalar el idProducto del dataset de accesorio, jalar el idCaja del dataset de caja y idClinica del dataset de caja.
16. **VenderMedicamento**
Mismo caso que *VenderJuguete*, así que es análogo.
17. **VenderComida**
Mismo caso que las tablas anteriores.
18. **ClienteNormal**
De manera similar a *supervisor* creamos un esquema auxiliar para la correcta generación de los curp de cada cliente. Generamos su dataset

Luego creamos un esquema con los respectivos campos en orden y jalamos el curp del esquema anterior. Para las llaves foráneas solo las jalamos de idCaja e idClinica del esquema de Caja.
19. **ClienteFrecuente**
Lo mismo que *ClienteNormal* sólo que éste tiene un atributo de más email, mockaroo ya nos permite agregar un email automáticamente.
20. **Fisico**
Creamos un esquema donde el id es generado por una expresión regular donde los cuatro primeros caracteres son números y los siguientes cuatro son letras.

Para las llaves foráneas, jalamos los curp de los dataset de clienteNormal y clienteFrecuente respectivamente.
21. **Internet**
Creamos un esquema donde el número de tarjeta es generado aleatoriamente con 16 dígitos, la creación de llaves foráneas es análogo a **Fisico**. El cvv es generado aleatoriamente por 4 dígitos, todos los demás atributos mockaroo los puede generar automáticamente.
22. **Veterinario**
Siguiendo el mismo principio de *Supervisor* creamos un esquema auxiliar para la correcta generación del curp y rfc.

Una vez teniendo el dataset creamos otro esquema para poner los datos en orden, el idClinica lo jalamos del campo idClinica del data set de Clinica. Todos los demás campos mockaroo permite generarlos sin ningún problema.
23. **Estilista**
De la misma manera que *Veterianrio* pero con sus respectivos campos.
24. **Mascota**
Creamos el esquema con sus respectivos campos de la tabla Mascota, el id es una expresión regular donde los 3 primeros caracteres son letras (las cuales hacen referencia a las primeras 3 letras del nombre de la mascota) y los últimos dos números son números generados al azar, jalamos el curp del cliente normal y curp del cliente frecuente de sus respectivos dataset.
25. **Normal**
Para la consulta normal generamos el id de manera secuencial con dos dígitos, para el idMascota

lo jalamos del dataset de Mascota así como el curp del dataset del Veterinario. Todos los demás campos se pueden generar con mockaroo.

26. **Emergencia**

Análogo que en el caso de *Normal*. En ésta tabla tenemos el campo codigoEmergencia para el cual ocupamos expresiones regulares de manera que las opciones son: verde, amarillo, rojo.

27. **Tratamiento**

Análogo a *Normal*.

28. **Recibo**

Creamos el esquema con sus respectivos campos de la tabla Recibo, el id es una expresión regular donde el primer caracter es una letra al azar entre N, E, T que significan normal, emergencia y tratamiento respectivamente; los 3 caracteres restantes son números al azar. Para las llaves foráneas jalamos del dataSet de Caja los atributos idCaja e idClinica. Para obtener los atributos referentes al nombre de la mascota y dueño de la mascota usamos el dataSet de mascota.

29. **GenerarReciboNormal**

Creamos un esquema donde usamos el dataSet de Recibo para tener el atributo idRecibo, debido a que es un recibo para una consulta normal usamos el dataSet de la consulta normal para el atributo idConsulta y idMascota. Los demás atributos se definieron usando las mismos tipos y expresiones que en las tablas Recibo y Normal.

30. **GenerarReciboEmergencia**

Es análogo que el caso anterior solo que ahora usamos el dataSet de consulta de emergencia para el atributo idConsulta y idMascota. Los demás atributos se definieron usando las mismos tipos y expresiones que en las tablas Recibo y Emergencia.

31. **GenerarReciboTratamiento**

Análogo a los dos casos anteriores cambiando el dataSet por el de tratamiento para el atributo idConsulta y idMascota. Los demás atributos se definieron usando las mismos tipos y expresiones que en las tablas Recibo y Tratamiento.

32. **Supervisor**

Practicamente es la mezcla entre los atributos de la tabla Supervisor y la tabla Clinica, al ser una relacion entre estos requiere de ambos datos, sus tipos y atributos al ser de estas dos mencionadas anteriormente son iguales

Cabe aclarar que para cada tabla generamos los datos y posteriormente los subimos como dataset para jalar información que necesitemos en otra tablas aunque esto no se diga explícitamente.

Consideraciones

Nos hubiera gustado que algunos datos tuvieran sentido, por ejemplo para las tablas que nos piden la descripción de un producto, motivo de consulta de la mascota, procedimiento emitido en las consultas, etc, hicimos uso del tipo "word" que nos ofrece Mockaroo el cual pone palabras al azar y con eso simulamos oraciones para llenar cada atributo en dichas tablas.

Para el nombre de las mascotas decidimos usar el tipo "username" ya que si usabamos "first name" había mascotas que tenían el mismo nombre que sus dueños o trabajadores de la clínica (personalmente no queríamos eso). También intentamos poner varios nombres de mascotas creados por nosotros al azar pero nos causó problemas en su id ya que había nombres que se repetían muy seguido.