

# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS.  
FUNDAMENTOS DE BASES DE DATOS.

## PRÁCTICA 07 POBLACIÓN DE BASE DE DATOS.

---

### EQUIPO: CHIQUESSQL

IVANA IX CHEL BONILLA NEGRETE  
315131994

DYLAN ENRIQUE JUAREZ MARTINEZ  
422117180

DANIEL ROJO MATA  
314297967

---

### PROFESOR:

GERARDO ÁVILES ROSAS

---

### AYUDANTES DE TEORÍA:

GERARDO URIEL SOTO MIRANDA  
VALERIA FERNANDA MANJARREZ ANGELES

---

### AYUDANTES DE LABORATORIO:

RICARDO BADILLO MACÍAS  
ROCÍO AYLIN HUERTA GONZÁLEZ

## I. ACTIVIDADES

1. Deberán crear un reporte en formato PDF en donde señalen la herramienta que utilizaron para la generación de datos (Mockaroo) y cómo fue que la utilizaron para generar los archivos que posteriormente cargaron a la base de datos. Y se llamará Práctica07.pdf.

Durante la población de la base de datos se utilizó la página: <https://www.mockaroo.com/>.

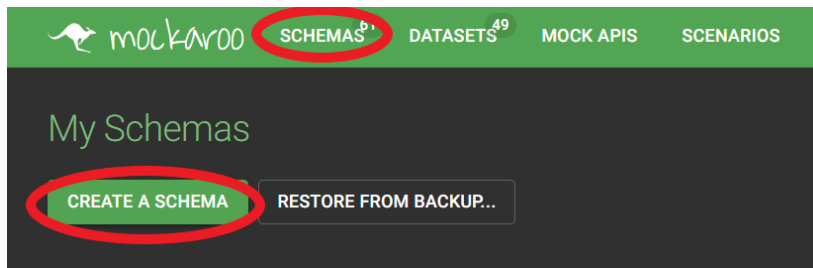
A continuación se explicarán los puntos más importantes del proceso. Ejemplificaremos dos casos: el primero es el poblamiento de la tabla **hotel**, la cual no cuenta con llaves foráneas, y el segundo caso, la tabla **telefonoHotel** la cual sí tiene.

### I.1. Tabla hotel

Primero evaluamos los campos a considerar, el tipo de datos y sus respectivos CONSTRAINT.

```
-- Tabla hotel
CREATE TABLE hotel (
  idHotel VARCHAR(20),
  nombreEstablecimiento VARCHAR(50),
  horaCheckIn TIME,
  horaCheckOut TIME,
  petFriendly CHAR(2),
  servicio VARCHAR(100),
  numeroInterior VARCHAR(10),
  numeroExterior INT,
  colonia VARCHAR(60),
  calle VARCHAR(70),
  estado VARCHAR(50)
);
```

Luego, volvimos a la página de Mockaroo y seleccionamos en la parte superior izquierda **SCHEMAS** seguida de **CREATE A SCHEMA**.



Esto nos llevó a la siguiente página, donde creamos los campos necesarios para cada elemento de la tabla. Observemos que se tomó el nombre idéntico de cada columna, además de que el tipo de dato se eligió considerando su compatibilidad con el dominio del atributo en la tabla.

En algunos casos, fue necesario señalar más especificaciones, tal es el caso del **numeroInterior** que no tiene una restricción de nulidad o el tipo *Regular Expression* de **nombreEstablecimiento** y **petFriendly**, en este último señalamos las distintas respuesta posibles en el apartado *Options*.

Field Name	Type	Options
idHotel	Row Number	blank: 0 % $\Sigma$ X
nombreEstablecimien	Regular Expression	(Hotel Luna) blank: 0 % $\Sigma$ X
horaCheckIn	Time	from: 3:00 PM to: 3:00 PM format: 24 Hour blank: 0 % $\Sigma$ X
horaCheckOut	Time	from: 1:00 PM to: 1:00 PM format: 24 Hour blank: 0 % $\Sigma$ X
petFriendly	Regular Expression	(SI NO) blank: 0 % $\Sigma$ X
servicio	Regular Expression	(Lavandería, Bar, Salones de eventos, Gimnasio Bar, Salones de eventos, Tienda de regalos, G blank: 0 % $\Sigma$ X
numeroInterior	Street Number	blank: 50 % $\Sigma$ X
numeroExterior	Street Number	blank: 0 % $\Sigma$ X
colonia	Plant Common Name	blank: 0 % $\Sigma$ X
calle	Street Name	blank: 0 % $\Sigma$ X
estado	State	Coahuila De Zaragoza Campeche Veracruz Llave Chiapas Puebla Chihuahua Aragon Baja California Guanajuato Nuevo Leon Morelos Tabasco Oaxaca Tamaulipas All Countries blank: 0 % $\Sigma$ X

Una vez que rellenamos todos los campos procedimos a elegir el número de filas y generamos el archivo en formato CSV.

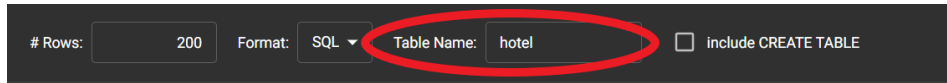
+ ADD ANOTHER FIELD
GENERATE FIELDS USING AI...

# Rows: 200
Format: CSV
Line Ending: Windows (CRLF)
Include: ☒ header ☐ BOM

Si bien pudo generarse el archivo SQL directamente se eligió de esta manera para guardar todos los datos en formato CSV (ya que por las especificaciones dadas mockaroo genera datos nuevos cada vez). Finalmente, se cargó el formato CSV creado en el dataset de Mockaroo y se utilizó en un nuevo schema:

Field Name	Type	Options
idHotel	Dataset Column	hotel idHotel sequential blank: 0 % $\Sigma$ X
nombreEstablecimien	Dataset Column	hotel nombreEstablecimiento sequential blank: 0 % $\Sigma$ X
horaCheckIn	Dataset Column	hotel horaCheckIn sequential blank: 0 % $\Sigma$ X
horaCheckOut	Dataset Column	hotel horaCheckOut sequential blank: 0 % $\Sigma$ X
petFriendly	Dataset Column	hotel petFriendly sequential blank: 0 % $\Sigma$ X
servicio	Dataset Column	hotel servicio sequential blank: 0 % $\Sigma$ X
numeroInterior	Dataset Column	hotel numeroInterior sequential blank: 50 % $\Sigma$ X
numeroExterior	Dataset Column	hotel numeroExterior sequential blank: 0 % $\Sigma$ X
colonia	Dataset Column	hotel colonia sequential blank: 0 % $\Sigma$ X
calle	Dataset Column	hotel calle sequential blank: 0 % $\Sigma$ X
estado	Dataset Column	hotel estado sequential blank: 0 % $\Sigma$ X

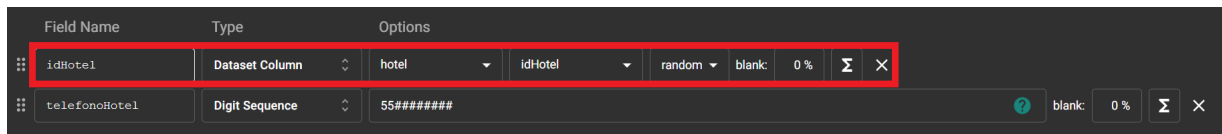
Notemos que se debe crear el archivo utilizando el nombre de la tabla utilizado en el DDL para evitar problemas en la ejecución del código al cargarlo en nuestra base de datos.



# Rows: 200 Format: SQL Table Name: hotel ☐ include CREATE TABLE

## I.2. Tabla telefonoHotel

El procedimiento realizado fue análogo al anterior con la única excepción de que en la creación de los datos para el archivo csv se utilizó el dataset de la tabla hotel para reutilizar lo idHotel generados que telefonoHotel utiliza como llave foránea.



Field Name	Type	Options
idHotel	Dataset Column	hotel id-Hotel random blank 0 % Σ X
telefonoHotel	Digit Sequence	55#####

## II. CORRECCIONES

DDL:

- Se borró la columna RFC que no correspondía en las habitaciones (Individual, Doble, Penthouse, Cuadruple).
- Corrección deL CONSTRAINT en la tabla penthouse para el número de camas. Se modificó para que pueda tener de una cama o más.