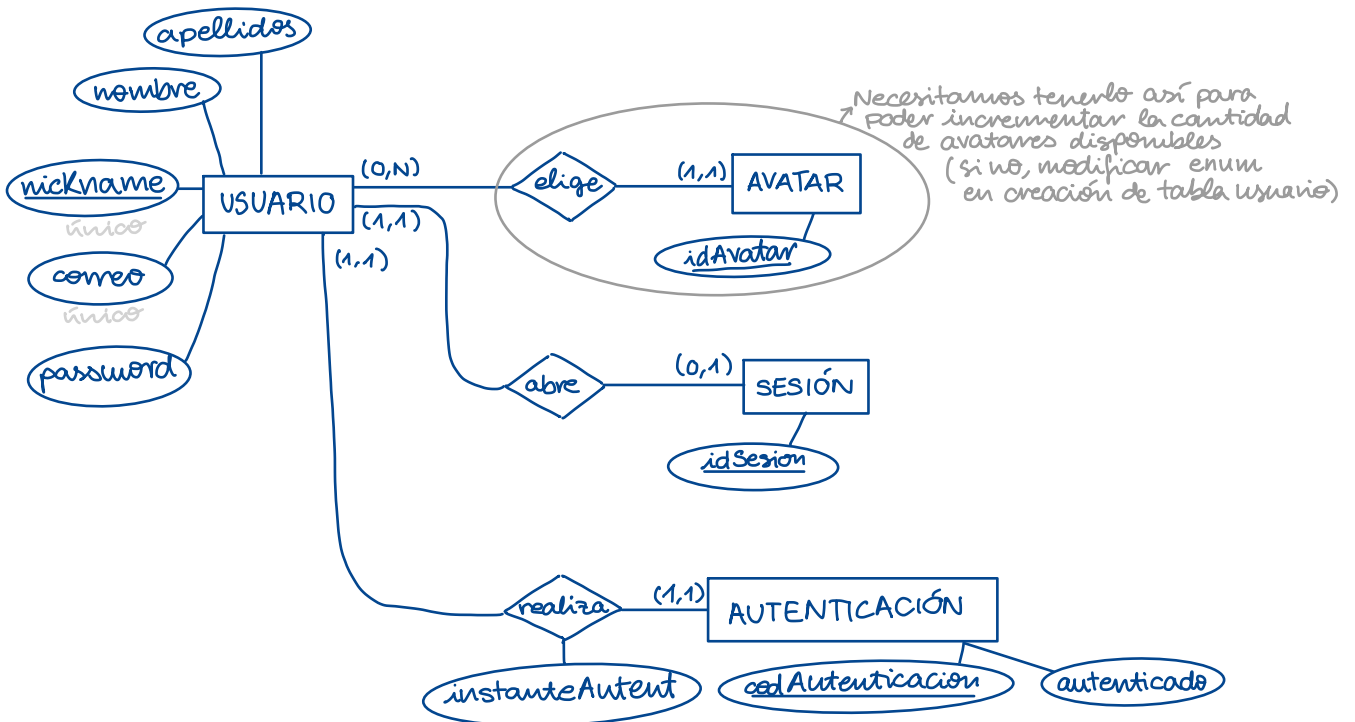


Dogmunity - Diseño de la BD

DER:



PASO A TABLAS:

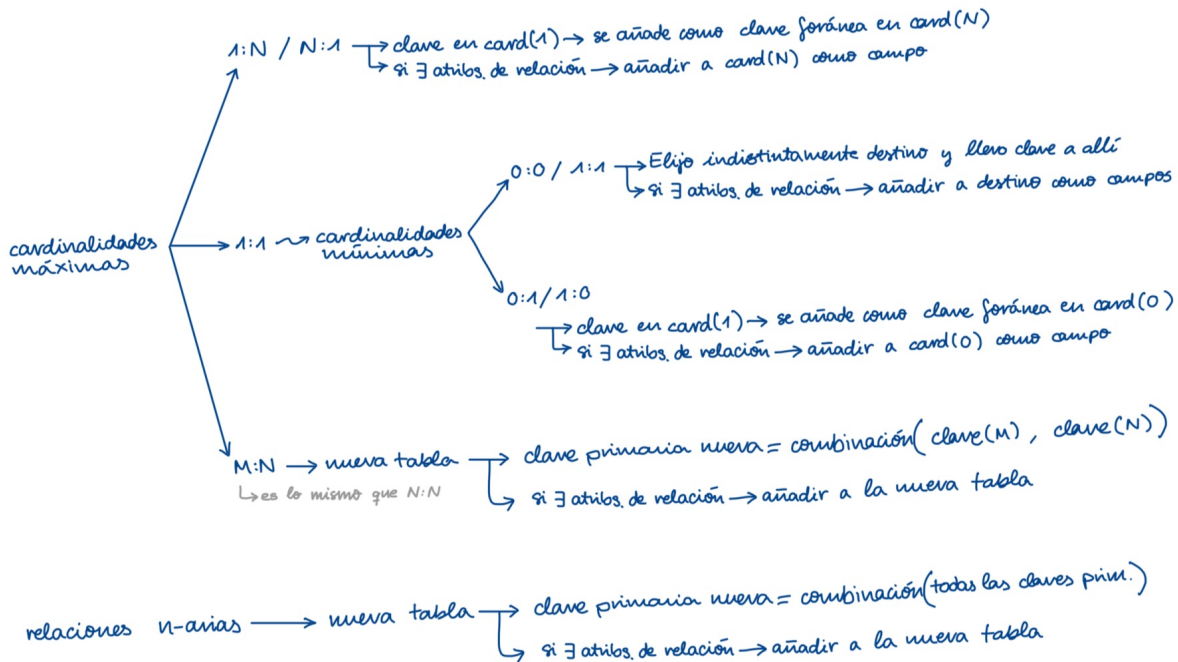
USUARIO(nickname, correo, password, nombre, apellidos, idAvatar)
 AVATAR(idAvatar)
 SESIÓN(idSesión, nickname)
 AUTENTICACIÓN(cdAutenticación, nickname, instanteAutent, autenticado)

TIPOS DE DATOS:

USUARIO	<ul style="list-style-type: none"> nickname PRIMARY KEY VARCHAR(30) correo UNIQUE NOT NULL VARCHAR(255) password NOT NULL VARCHAR(100) → a futuro: guardarla encriptada nombre VARCHAR(50) NOT NULL apellidos VARCHAR(100) NOT NULL idAvatar TINYINT UNSIGNED → Cada foto de Avatar la numeramos. 1, 2, 3, ...
AVATAR	<ul style="list-style-type: none"> idAvatar PRIMARY KEY TINYINT UNSIGNED
SESIÓN	<ul style="list-style-type: none"> idSesión PRIMARY KEY VARCHAR(255) nickname VARCHAR(30) UNIQUE NOT NULL
AUTENTICACIÓN	<ul style="list-style-type: none"> cdAutenticación PRIMARY KEY CHAR(6) → típica de 6 valores. nickname VARCHAR(30) UNIQUE NOT NULL instanteAutent TIMESTAMP NOT NULL autenticado ENUM('TRUE', 'FALSE') NOT NULL DEFAULT 'FALSE'

Avatar por defecto

PASO A TABLAS



Volcado lógico usando MySQL:

Métodos de backup de una base de datos

17/04/2021

Backups con mysqldump

- mysqldump** es un programa externo que permite volcar la estructura y/o contenido de bases de datos MySQL en un **fichero de texto plano** → el más compatible
- Este fichero de volcado puede usarse:
 - Como **backup** para permitir la recuperación en caso de pérdida de datos
 - Como fuente de datos para **configurar esclavos de replicación**
 - Como fuente de datos para **experimentación** (copias de la base de datos)
- Produce 2 tipos de salidas, dependiendo del parámetro **--tab**:
 - Sin --tab**: se crea un único fichero de salida (*.sql) conteniendo **sentencias SQL** (CREATE, INSERT...)
 - Con --tab**: se crean dos ficheros por cada tabla volcada. Uno denominado **tbl_name.txt** conteniendo los **datos** de la tabla, separados por tabuladores, y otro **tbl_name.sql** con la sentencia de **creación de la tabla** (CREATE TABLE)
- Información completa: <http://dev.mysql.com/doc/refman/8.0/en/mysqldump.html>

conclusión: sirve para duplicar los datos en la propia máquina del servidor o en otra máquina



16

Métodos de backup de una base de datos

<https://dev.mysql.com/doc/refman/8.0/en/mysqldump-sql-format.html>
<https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html>

Backups con mysqldump (II)

- Por defecto, **mysqldump** escribe la información por la salida estándar. Es necesario redirigirlo a un fichero para guardar el resultado

shell> **mysqldump** [credenciales -h-u-] [dirección de la salida] [arguments] [file_name] [debe tener permiso]

- Para volcar **todas** las bases de datos: (la pantalla)

shell> **mysqldump --all-databases > dump.sql**

- Para volcar **bases de datos específicas**:

shell> **mysqldump --databases db1 db2 db3 > dump.sql**

En estos dos últimos casos, **mysqldump** escribirá sentencias de **CREATE DATABASE** y **USE** en el fichero de salida. Se puede añadir **--add-drop-database**



17

- Volcar **por separado estructura y datos**:

```
shell> mysqldump --no-data db1 > dump-defs.sql → ESTRUCTURAS
shell> mysqldump --no-create-info db1 > dump-data.sql → DATOS (inserts)

shell> mysqldump --no-data --routines --events db1 > dump-defs.sql
```

sql Copiar código

```
CREATE TABLE clientes (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL
);
```

2. Tabla de Pedidos con clave foránea:

sql Copiar código

```
CREATE TABLE pedidos (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  id_cliente INT UNSIGNED,
  fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  CONSTRAINT fk_cliente FOREIGN KEY (id_cliente) REFERENCES clientes(id)
);
```

Consideraciones:

1. Integridad Referencial: La clave foránea garantiza que no puedes insertar un `id_cliente` en la tabla de `pedidos` que no exista en la tabla de `clientes`. Si intentas hacerlo, obtendrás un error.

→ cómo poner foreign Keys en MySQL