**Entidades & Atributos.**

**Sobre los principales objetos de manipulación dentro de una base de datos relacional.**







**Pasos para crear una base de datos relacional en nuestro proyecto (esta vez el proyecto será un sistema de "Platziblog").**

**En un proyecto real, para crear una base de datos relacional, lo primero que hay que hacer es:**

**1. Identificar las entidades (para nuestro caso, serían las entidades de un “Blog” de Platzi).**

*1ra entidad:* ***"Posts":*** *El cuerpo propiamente del Blog.*

*2da entidad:* ***"Usuarios":*** *Los que van a crear el contenido del Blog.*

*3ra entidad:* ***"Comentarios":*** *Caja de comentarios al contenido del Blog.*

*4ta entidad:* ***"Categorias":*** *Tienen que ver con las sesiones del "blog"; es decir, las pestañas que tiene el blog por separado por cada tema de interés. Las categorias pueden ser: "Política", "Deportes", "Economía", "Moda", "Tecnología", etc.*

**2. Identificar los atributos (para nuestro caso, serían los atributos de un “Blog” de Platzi).**

***Para la entidad "Posts":***

*1er atributo:* ***Un título.***

*2do atributo:* ***La fecha\_de\_publicación.***

*3er atributo:* ***el Contenido.***

*4to atributo:* ***El estatus*** *(si fue publicado o no, si está activo o no).*

*5to atributo:* ***Las etiquetas****, las cuales son Atributos Multivaluados; es decir, hay varias etiquetas realmente y se encierran en un ovalo con doble subrayado. La etiqueta es la que nos dicta que tipo de post estás leyendo; es decir, si hace parte de la sesión de "ciencias", "tecnología", "deportes", etc. O bien, en un mismo post, pueden haber varias sesiones; en ese caso, serían varias etiquetas (una por sesión), es por lo anterior que en el atributo "etiqueta" hablamos de que sea una etiqueta "multivalor".*

*6to atributo:* ***Un id, esta sería la etiqueta "llave" o clave de nuestra entidad "Posts"****; es lo que nos dirá que todos los* ***posts*** *serán diferentes entre sí, al menos, minimamente, por el "id".*

***Para la entidad "Usuarios":***

*1er atributo:* ***Un “login”****, para poder hacer inicio de sesión con una cuenta.*

*2do atributo:* ***Un “password”****, para tener acceso a nuestra cuenta con una contraseña.*

*3er atributo:* ***Un apodo o “alias”.***

*4to atributo:* ***Un email.***

*5to atributo:* ***Un id, esta sería la etiqueta "llave" o clave de nuestra entidad "Usuarios"****; es lo que nos dirá que todos los* ***usuarios*** *serán diferentes entre sí, al menos, minimamente, por el "id".*

***Sobre las relaciones***

**Las relaciones es la manera en la que empezamos a ligar nuestras diferentes entidades u objetos (De aquí surge la importancia de los Diagramas entidad-relación)**

Las relaciones se representan con un **Rombo.**

**Cómo funcionan las relaciones?**

Supongamos que tenemos la entidad *Automovil* y la entidad *Dueño*, siendo en total dos (2) entidades:

Automovil

Dueño

Ahora, lo que nos permite vincular o relacionar ambas entidades son las ***relaciones,*** *éstas por convención se definen por medio de* ***verbos.*** En este caso, un verbo más apropiado de la relación que hay entre las entidades “**automovil” y “dueño” *es “*tener”.** Quedando visualmente este vinculo de entidades y relaciones así:

Automoviles

tienenee

Dueños

Es necesario el uso de *líneas conectoras* para poder ligar o hacer la unión entre las entidades, a esta representación visual se le llama: **“Diagramas entidad-relación”**.

***Otro ejemplo de relación entre dos entidades podría ser:***

Jugadores

pertenecen

Equipos

Lo que distingue una entidad de un atributo es que el atributo es algo propio a una entidad; es decir, es una caracteristica de la entidad u objeto; mas, cuando se habla de dos entidades, se habla de dos objetos por separado y completamente independientes entre sí. **Sin embargo, hay un tipo de atributo en especifico que recibe un tratamiento igual al de una entidad en términos de *relaciones, y son: Los atributos Multivaludos.*** Veamos esto en un caso que ya hemos trabajado.

Laptops

tienen

discos\_duros

Si se da cuenta, realmente los *discos duros* no son independientes a una Laptop; antes, son un elemento o caracteristica propia de ellas (de las laptops); es decir, los *discos duros* no son una entidad como tal; sin embargo, como es un *atributo multivaluado*, recibe el mismo tratamiento en términos de relaciones que cualquier otra entidad (convertiendose, en ese sentido, en una entidad más). Lo anterior se debe a que los atributos multivaluados son más complejos y pueden relacionarse de varias formas con su entidad.

Ahora, para definir o saber puntualmente la cantidad de, por ejemplo, los *discos duros* que tiene una *laptop* se hace por medio de una nueva propiedad llamada: ***Cardinalidad;*** es decir, la ***cardinalidad*** vendría siendo lo que nos permite saber la cantidad de entidades o de atributos multivaluados que hay de lado y lado en cada relación resuelta. O también se puede definir la ***Cardinalidad*** como: **La facultad de saber la cantidad numérica de valores que tiene un atributo multivaluado (o una entidad -otra) para una entidad en cuestión (al ser relacionadas ambas)”.**

***Cardinalidad: casos o diferentes tipos.***

***Cardinalidad: 1 a 1***



Cuando se habla de ***cardinalidad: 1 a 1,*** nos referimos a que sólo hay un valor, para lado y lado, en la relación etiqueta-etiqueta o etiqueta-atributo multivaluado; en nuestro ejemplo, para la entidad “persona”, nos estaríamos refiriendo a una sola persona como tal por separado; por otra parte, para el atributo multivaluado “datos-contacto”, nos referimos a los datos de contacto correspondientes a una única persona precisamente. ***Esta representación gráfica reflejada es un diagrama entidad-relación.*** Sin embargo, existe otro tipo de diagrama para representar visualmente la misma información dada por un caso de cardinalidad: 1 a 1,... y son los ***Conectores*** *de un diagrama fisico de una base de datos****.***



El primer conector básicamente nos dice lo mismo: *“de un lado tengo uno y del otro lado tengo otro”;* mas, el segundo conector también significa lo mismo, sólo que refuerza la declaración de que en un lado efectivamente tengo uno *(y sólo uno)* y del otro lado tengo uno también *(y sólo uno).* Esa expresión de *“sólo uno”* se simboliza gráficamente con la doble raya a los extremos del conector.

***Cardinalidad: 0 a 1 (o Cardinalidad: 1 a 1 opcional)***

Acá puede haber la opción de que no exista ninguno de uno de los dos lados. Veamos el siguiente caso.



Si analizamos la “sesión\_actual”, la sesión de un usuario, con el “usuario” mismo; podemos decir que toda “sesión\_actual” tiene que tener un usuario, pero un usuario puede *no estar en sesión* en determinado momento; por lo tanto, la *cardinalidad se define como de: 0 a 1,* pues, puede que no haya una *sesión actual.*

***Cardinalidad: 1 a N (o 1 a muchos)***

***La Cardinalidad: 1 a muchos*** quiere decir, como su nombre indica, que de un lado tenemos uno pero del otro lado tenemos muchos. Veamos el siguiente ejemplo.



Como logra observar, se plantea que una persona puede tener muchos automoviles.

***Cardinalidad: 0 a N (o 0 a muchos)***

Acá puede haber la opción de que no exista ninguno de uno de los dos lados, pero muchos en el otro lado restante. Veamos el siguiente caso.



Si analizamos al “paciente”, el que ocuparía la habitación de hospital, con la “hab\_hospital” misma; podemos decir que todo “paciente” tiene que tener asignado una “hab\_hospital”; pero, en algún caso, muchas de esas *habitaciones de hospital* quedan completamente desocupadas de *pacientes;* por lo tanto, la *cardinalidad se define como de: 0 a N,* pues, puede que no haya ningún *paciente* para una cantidad considerable de *habitaciones de hospital* disponibles.

***Cardinalidad: N a N (muchos a muchos)***

Esta es, quizás, el tipo de *cardinalidad* más especial; se separa un poco del resto de *cardinalidades* porque no contempla a una “*entidad principal”* con un único valor posible. *Generalmente son muchos de lado y lado*. Veamos el siguiente ejemplo.



Un alumno puede estar inscrito o tomar varias clases; puede, por ejemplo, tomar Español, Matematicas o Fundamentos de bases de datos (1-N); pero adicionalmente una clase, cualquiera de esas tres mencionadas, puede estar *relacionada* o contiene a varios alumnos (N-1). El último conector de esta cardinalidad es más extricto que el primero; pues, nos cuenta que, en todos los casos, siempre habrá *muchos* de un lado y *muchos* de otro: no puede haber *0s,* no pueden haber *1s*, siempre habrán *muchos.* Ahora, lo más común es que se dé el primer conector y no el último entre la *cardinalidad: N a N.*