

Tarea 3. Modelo Relacional.

1. Preguntas de repaso

- a. ¿Qué es una **relación** y qué características tiene?

Una relación (relationship) es una asociación entre entidades, matemáticamente se refiere a:
Si A, B son conjuntos, entonces una relación R es un subconjunto de $A \times B$

- b. ¿Qué es un **esquema de relación**?

Es un conjunto finito de relaciones junto con una serie de restricciones o reglas de integridad.

- c. ¿Qué es una **llave primaria**?, ¿qué es una **llave candidata**?, ¿qué es una **llave mínima**?, ¿qué es una **super llave**?

Una llave primaria es un conjunto no vacío de atributos que identifican de manera única a cada tupla.

Sea R una relación, entonces una **llave candidata** para R es un subconjunto del conjunto de atributos de R, digamos K tal que:

- **Unicidad.** No existe par de tuplas distintas en R con el mismo valor para K.
- **Irreductibilidad.** Ningún subconjunto propio de K tiene la propiedad de unicidad..

Por otra parte una **super llave** es cualquier conjunto de atributos que identificaran un elemento, es decir, un registro. Mientras una **llave mínima** es una llave candidata con el menor número de elementos posibles.

- d. ¿Qué restricciones impone una **llave primaria** y una **llave foránea** al modelo de datos relacional?

Una **llave primaria** jamás existe con valor nulo.

Cuando una llave c en una relación R aparece como atributo en otra relación S, se dice que c es una **llave externa** en S.

Sea R2 una relación, se define como una **llave externa** en R2 como un subconjunto FK, del conjunto de atributos de R", tales que:

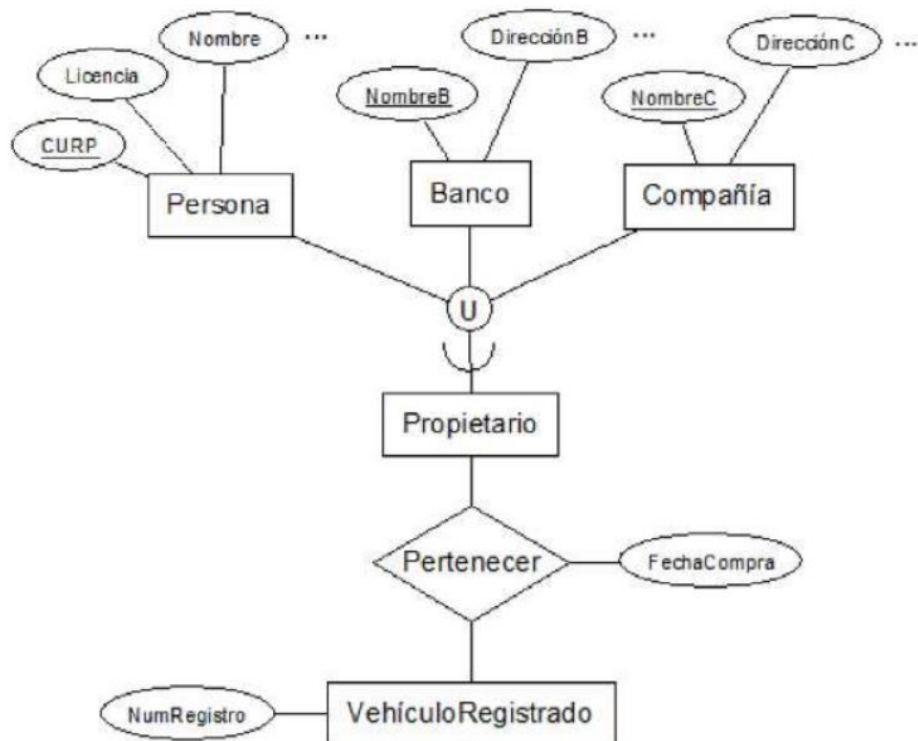
- Existe una relación R1 con una llave FK, y
- Cada valor de FK en R2 es idéntico al valor de FK en alguna tupla de R1.

- e. Investiga cómo se traducen las **categorías** (presentes en el modelo E/R) al **modelo relacional**. Proporciona un ejemplo.

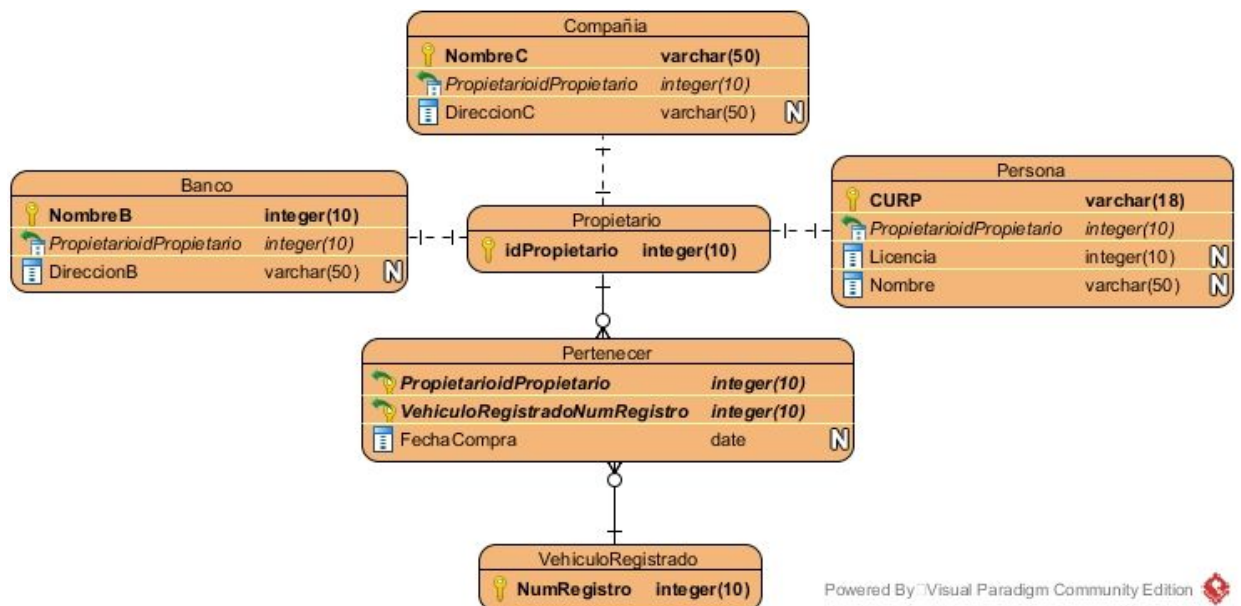
Para traducir las categorías tenemos dos posibles casos:

- Si las superclases de la categoría tienen diferentes llaves primarias:
 - Se crea una relación R que corresponda a la categoría y se le asigna una llave sustituta arbitraria.
 - Se añade a la llave sustituta a modo de llave foránea en cada una de las relaciones Ri que corresponden a las superclases de la categoría.
- Si las superclases de la categoría tienen la misma llave primaria:
 - Se crea una relación R que corresponda a la categoría y se le asigna como atributo llave primaria la llave común a todas las superclases de la categoría.

Para el ejemplo se usó un diagrama visto en clase pero no se tradujo al modelo relacional.



La traducción correspondiente es la siguiente:



2. Modelo relacional

Traduce el siguiente modelo **Entidad/Relación** a su correspondiente **Modelo Relacional**:

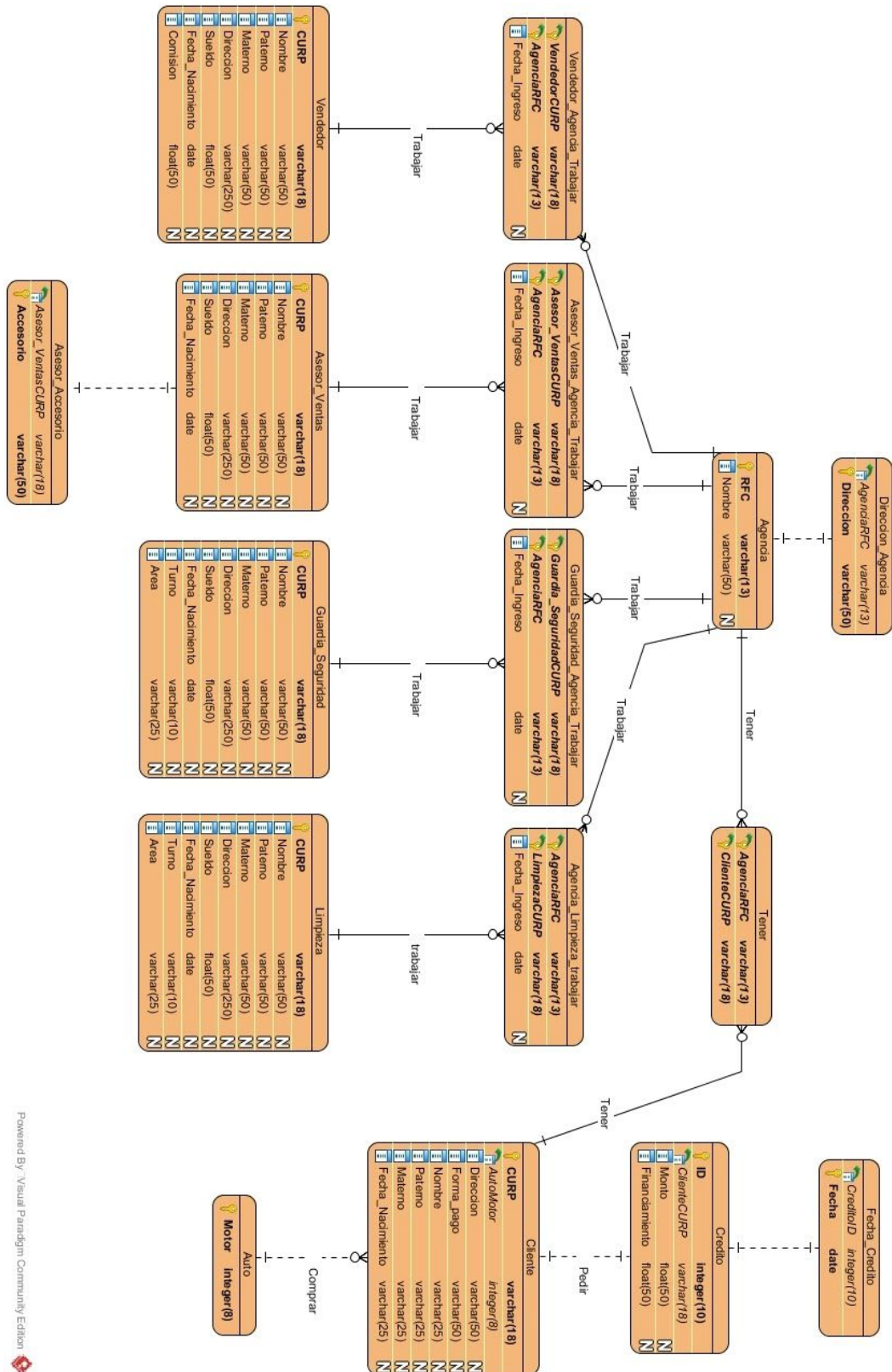
3. Modelo relacional

Traduce a su correspondiente **Modelo Relacional**, los problemas de la **venta de autos** y el **juego de computadora** (ambos de la **Tarea 1**). Si realizaste alguna modificación a tu diseño original (para mejorarlos), indica los cambios hechos y la justificación de los mismos.

En cualquier caso, deberás mostrar el **diagrama E/R** y su correspondiente traducción. Es importante que muestres tanto las **restricciones de entidad** como las de **integridad referencial**.

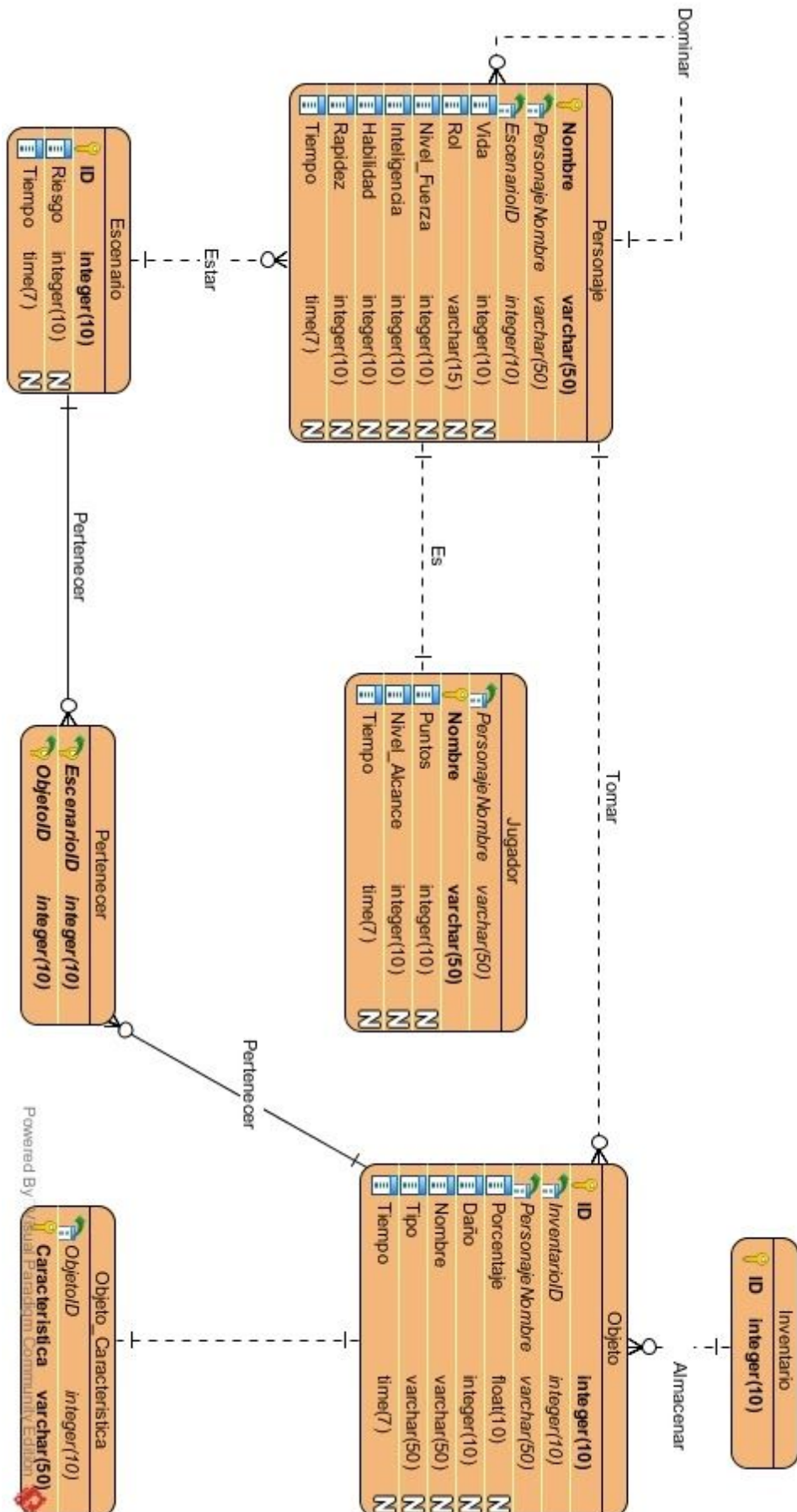
- Problema de la **venta de autos**

Se cambió la cardinalidad del diagrama de la tarea anterior, ya que un cliente puede comprar muchos autos.



- Problema de **juego de computadora**.

Se le cambio el atributo tiempo de compuesto a simple, para poder usar el tipo time.



4. Lectura

Leer el artículo **Codd's 12 Rules for an RDBMS**. Realiza un breve resumen y explica con tus propias palabras cada una de las 12 reglas de **Codd**.

Indica por qué consideras que es importante y si, hasta el momento de los comentado en el curso sería posible que un **SMBD** pudiera cumplir enteramente con lo que ahí se propone.

Tienen el objetivo de normalizar la definición de las bases de datos relacionales, ya que muchas veces no se sigue el modelo relacional estrictamente. Además si sería posible que un SMBD pudiera cumplir enteramente con las reglas que se propone, pero algunas de ellas son difíciles de realizar. A continuación se da una breve explicación de cada una de estas reglas.

1. La regla de la información
Toda la información en la base de datos se representan por valores dentro de columnas y filas. Todos los valores tiene que ir dentro de una tabla. Toda los datos en la base de datos se representa en valores dentro de una tabla que tiene columnas y filas.
2. La regla del acceso garantizado
Los datos tienen que ser accesibles y sin malas interpretaciones, es decir, los datos deben de tener un identificador único para lograr acceder a ellos. En otras palabras, cada pieza atómica de dato es accesible por la combinación del nombre de la tabla, valor de la llave primaria y nombre de la columna.
3. Tratamiento sistemático de valores nulos.
Las base de datos deben manejar la lectura de datos nulos para lograr la representación de datos faltantes, inexistentes o no conocidos.
4. Catálogo dinámico en línea basado en el modelo relacional
Los usuarios autorizados deben poder ingresar al catálogo en línea(estructura).
5. La regla comprensiva del sublenguaje de los datos.
Se debe soportar al menos un lenguaje relacional.
 - Sintaxis lineal.
 - Acceso de forma interactivo y/o desde aplicaciones.
 - Operaciones de definiciones, manipulación, seguridad, y restricción de lo datos.
 - Administrar transacciones.
6. Regla de actualización
El sistema debe tener la capacidad de actualizar automáticamente sin la necesidad de hacerlo manualmente.
7. Alto nivel de inserción, actualización y borrado
La base de datos tiene que tener la capacidad de poder borrar, copiar o mover en grandes cantidades. Es decir, Permite al sistema realizar manipulación de datos de alto nivel, es decir, sobre conjuntos de tuplas. Esto significa que los datos no solo se pueden recuperar de una base de datos relacional de filas múltiples y/o de tablas múltiples, sino que también pueden realizarse inserciones, actualizaciones y borrados sobre varias tuplas y/o tablas al mismo tiempo.
8. Independencia física de los datos
Los cambios internos no afectan a la vista del usuario, es decir, las aplicaciones deben trabajar de la misma manera.
9. Independencia lógica de los datos
Las aplicaciones y actividades de un computador permanece inalterado a nivel lógico sin importar cuales sean los cambios que se realicen en las tablas que almacenan datos.
10. Independencia de la integridad
Una base de datos tiene que tener restricciones de integridad específicas:
 - Integridad de identidad: No se permite que ningún componente de una llave primaria tenga un valor NULL.

- Integridad referencial: Para cada valor distinto de la llave foránea diferente de NULL en una base de datos relacional, debe existir un valor de la llave primaria que coincida con el mismo dominio.

11. Independencia de la distribución

Las porciones de la base de datos cuando se deben o pueden repartir en varias localizaciones, esta debe de ser invisible para los usuarios y la base de datos debe seguir funcionando sin ningún problema en el caso de actualizar el DBMS, por dar un ejemplo.

12. La regla de no subversión.

Si el sistema proporciona una interfaz de bajo nivel de registro, que esa interfaz no se pueda utilizar para subvertir(cambiar o transformar) el sistema.