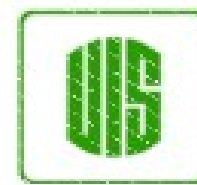
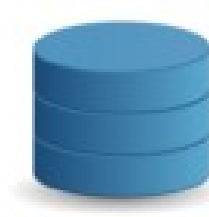


Modelo Relacional MR



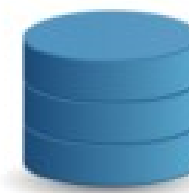


Modelo Relacional Elementos

El modelo debe estar compuesto por:

- **Tablas**
- **Atributos**
- **Llaves**
- **Relaciones**
- **Cardinalidad**

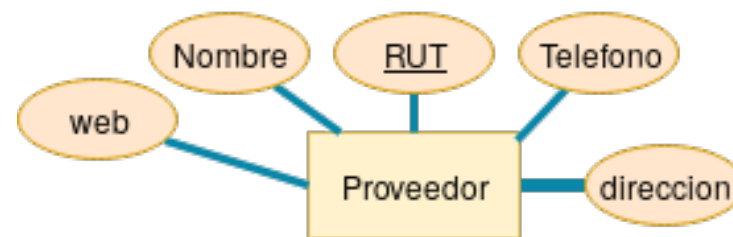
Este representa la base de datos por medio de tablas en las cuales se almacenará la información. El diagrama del modelo relacional se crea a partir del diagrama entidad relación siguiendo una serie de reglas.



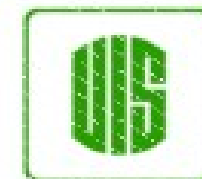
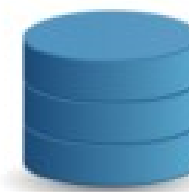
- **Modelo Relacional Elementos>>Campos**

Todas las entidades del modelo entidad relación se convierten en tablas en el modelo relacional, y sus atributos en campos.

Los campos son cada una de las columnas que forman la tabla. Ejemplo.



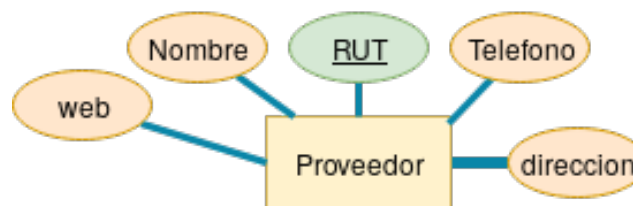
Proveedor
nombre
RUT
telefono
direccion
web



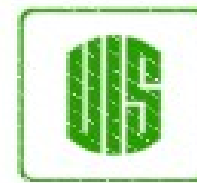
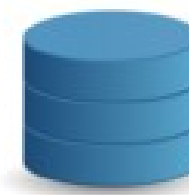
Modelo Relacional Elementos>>LLaves

- **PK (Primary Key):** El atributo principal de cada entidad en el modelo entidad relación lo llamaremos llave primaria.
Es un campo o una combinación de campos que identifica de forma única a cada fila de una tabla.

En el modelo relacional y por facilidad lo representaremos con las letras **PK (Primary Key)**, las cuales irán a continuación del nombre del campo.



Proveedor
PK RUT
nombre
telefono
direccion
web

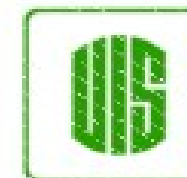
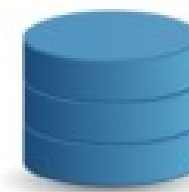


- **Modelo Relacional Elementos>>LLaves**

Llave foránea es un limitación referencial entre dos tablas, identifica una columna o grupo de columnas en una tabla (tabla hija) que se refiere a una columna o grupo de columnas en otra tabla (la tabla maestra).

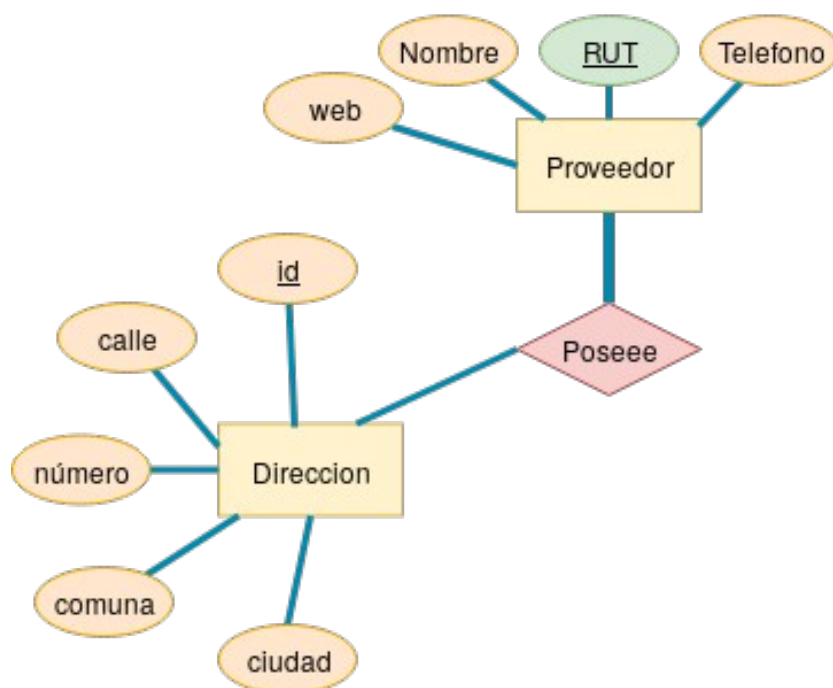
Las columnas en la tabla referente deben ser la clave primaria u otra clave candidata en la tabla referenciada.

La Llave Foránea también se determinar en el momento de las relaciones entre las tablas en la cual la Llave Primaria se realizar la unión con el otro campo de otra tabla y automáticamente se vuelve una llave foránea



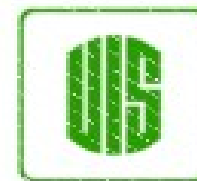
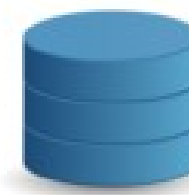
- Modelo Relacional Elementos>>LLaves

Llave foránea



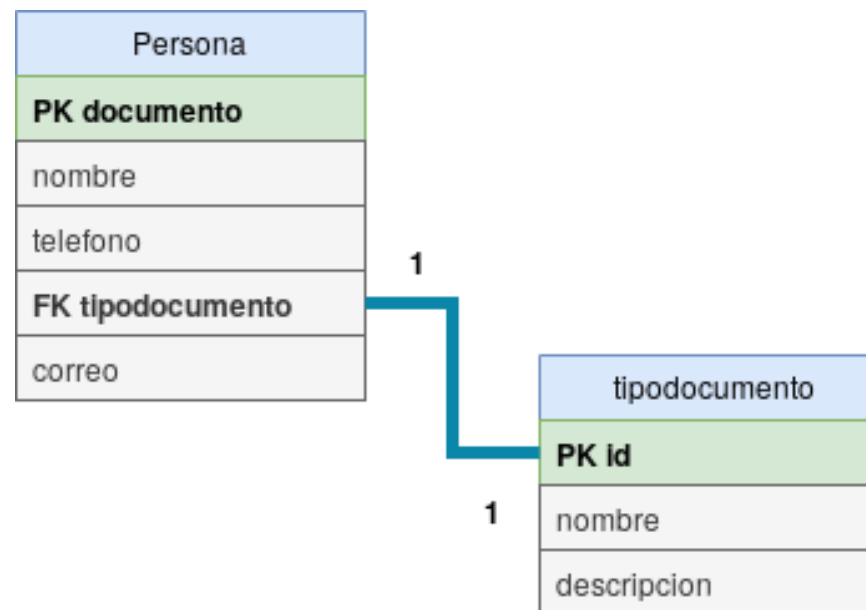
Proveedor
PK RUT
nombre
telefono
FK direccion
web

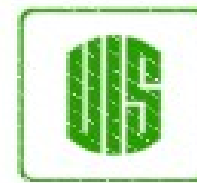
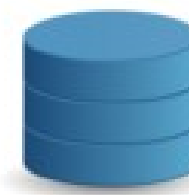
Direccion
PK id
calle
numero
comuna
ciudad



- **Modelo Realacional Elementos>>Relaciones**

- **Relacion 1:1** : Donde exista una relación Uno a Uno (1:1) entre 2 entidades, se debe crear una llave que llamamos foránea, en una de las 2 tablas asociadas.

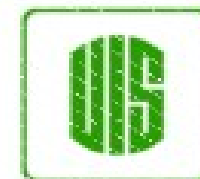
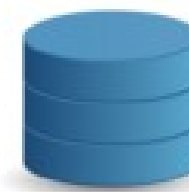




- **Modelo Realacional Elementos>>Relaciones**

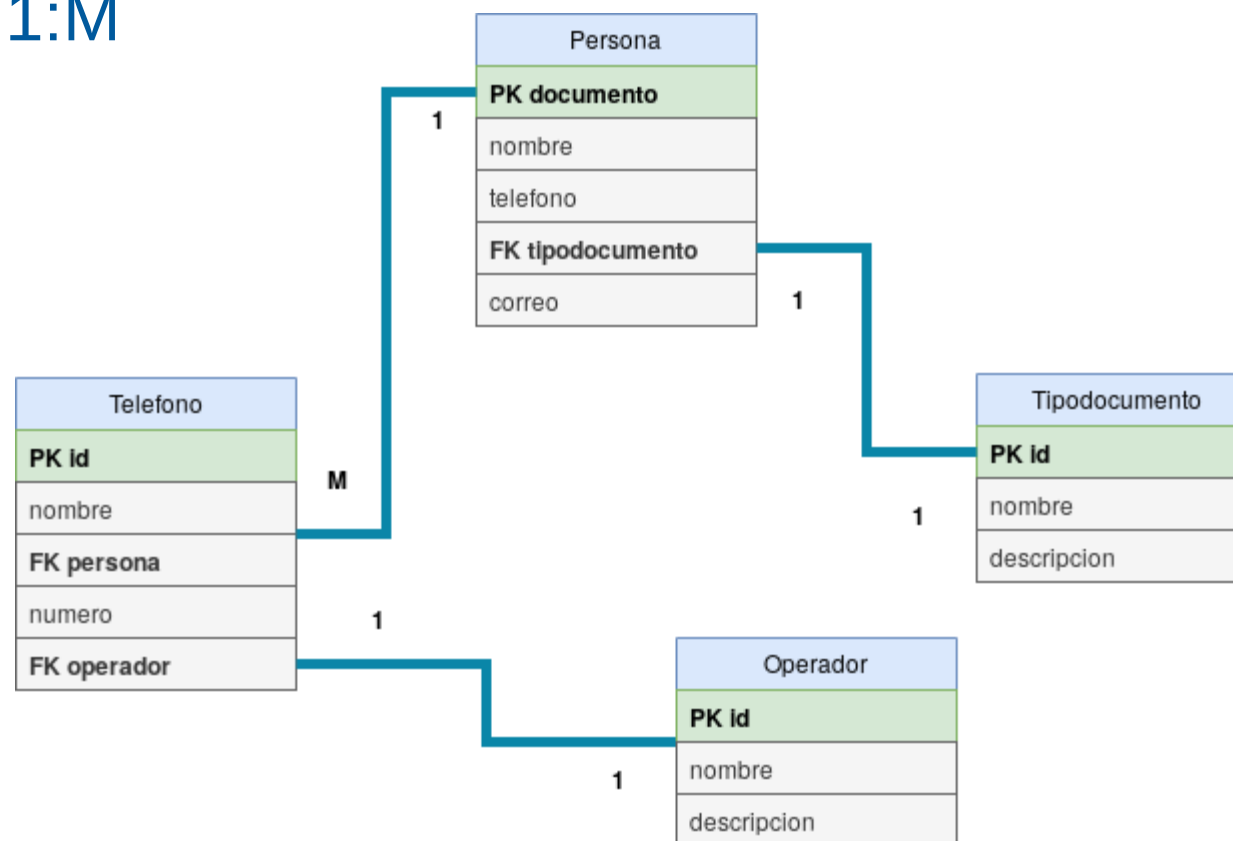
- **Relacion 1:M** :Donde exista una relación de uno a muchos (1:M) en el modelo entidad relación, crearemos las tablas correspondientes en el modelo relacional, y donde se encontraba la multiplicidad muchos (M) se creará obligatoriamente la llave foránea.

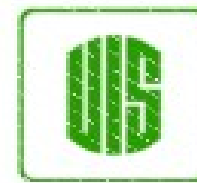
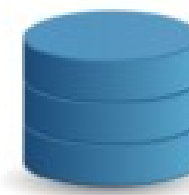
Ejemplo las entidades Persona y telefono. Una persona puede tener varios Telefonos, pero cada Telefono pertenece a una sola persona.



- **Modelo Relacional Elementos>>Relaciones**

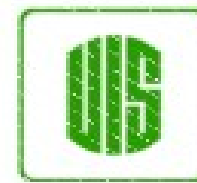
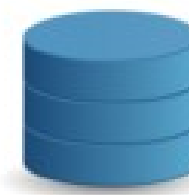
- **Relacion 1:M**





- **Modelo Relacional Elementos>>Relaciones**

- **Relacion M:M** : Donde exista una relación de muchos a muchos en el modelo entidad relación, la relación se convierte en una tabla nueva que une las 2 entidades que se estaban relacionando originalmente.
- La nueva tabla tendrá su propia llave primaria Es un campo o una combinación de campos que identifica de forma única a cada fila de una tabla. y tendrá 2 llaves foráneas, las cuales corresponden a las llaves primarias de las otras dos tablas.

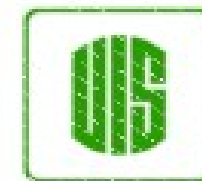
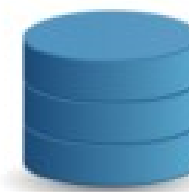


- **Modelo Realacional Elementos>>Relaciones**

- **Relacion M:M :**

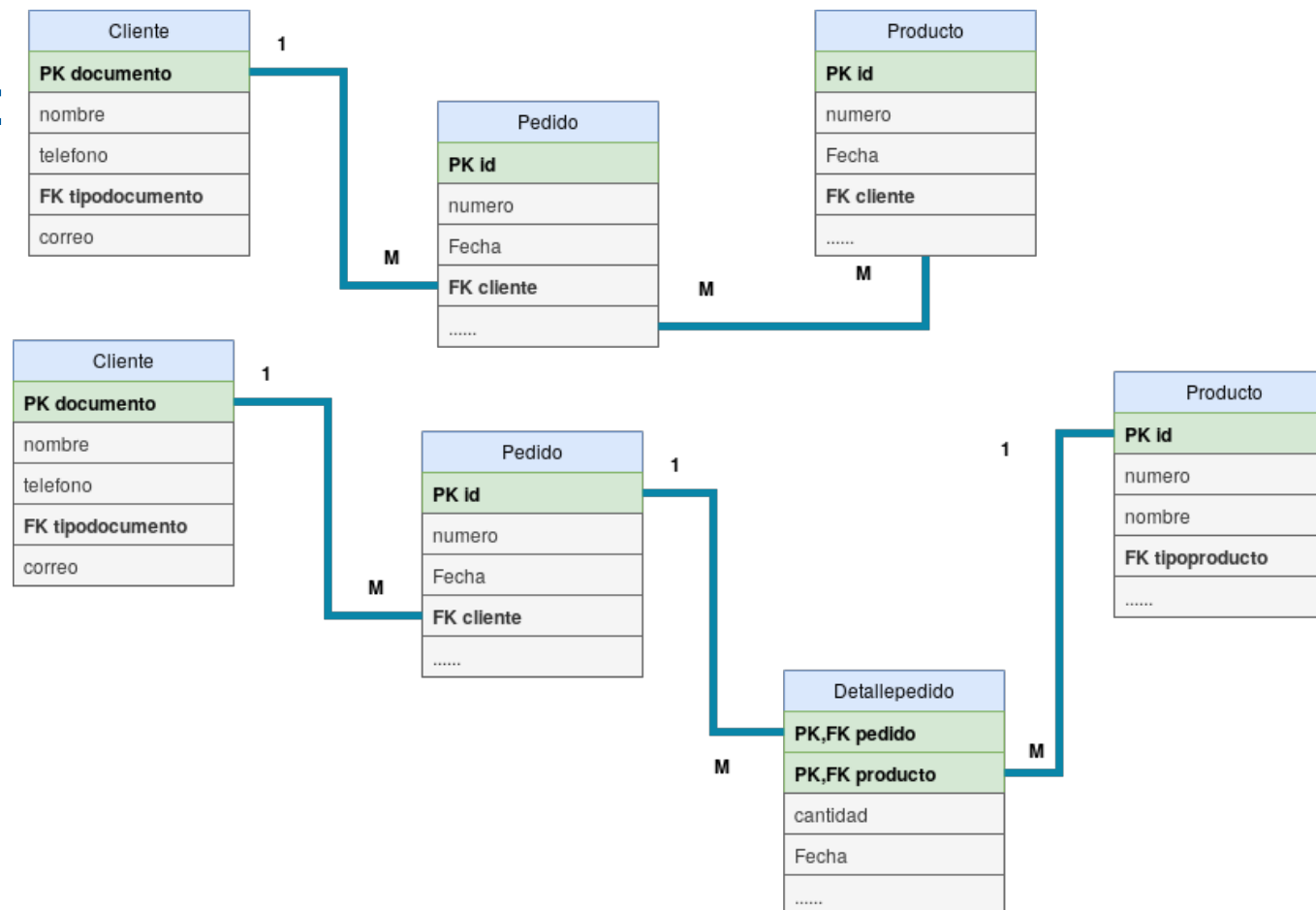
Adicionalmente tendrá los campos Los campos son cada una de las columnas que forman la tabla. adicionales que sean requeridos. El nombre para la nueva tabla puede ser el mismo de la relación u otro que sea más adecuado.

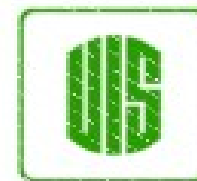
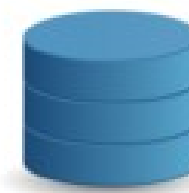
- Para la cardinalidad se mantiene la misma regla aplicada en los casos anteriores. Se colocará un 1 donde se encuentran las llaves primarias, y una M donde se encuentra las llaves foráneas, transformándose así en dos relaciones de uno a muchos (1:M).



• Modelo Relacional Elementos>>Relaciones

• Relacion M:M :



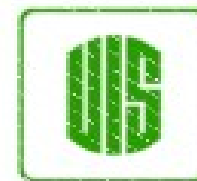
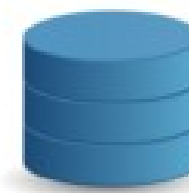


- **Modelo ER Elementos>>Ejemplo**

Sistema de ventas

Le contratan para hacer una BD que permita apoyar la gestión de un sistema de ventas. La empresa necesita llevar un control de proveedores, clientes, productos y ventas. Un proveedor tiene un RUT, nombre, dirección, teléfono y página web. Un cliente también tiene RUT, nombre, dirección, pero puede tener varios teléfonos de contacto.

La dirección se entiende por calle, número, comuna y ciudad. Un producto tiene un id único, nombre, precio actual, stock y nombre del proveedor. Además se organizan en categorías, y cada producto va sólo en una categoría. Una categoría tiene id, nombre y descripción. Por razones de contabilidad, se debe registrar la información de cada venta con un id, fecha, cliente, descuento y monto final. Además se debe guardar el precio al momento de la venta, la cantidad vendida y el monto total por el producto.

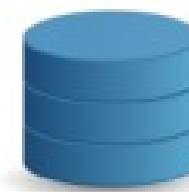


- **Modelo ER Elementos>>Ejemplo**

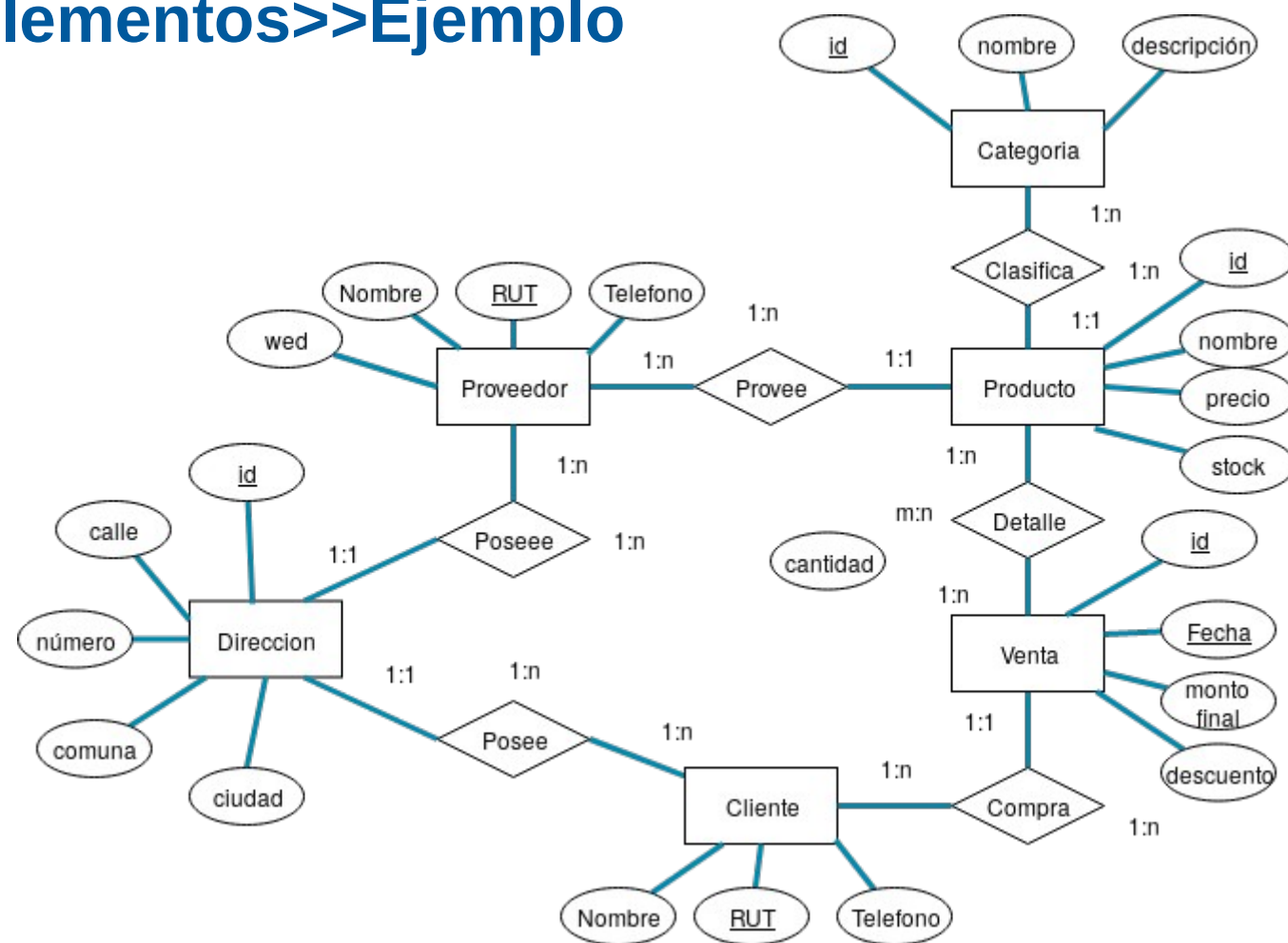
Sistema de ventas (identifiquemos Entidades)

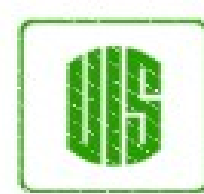
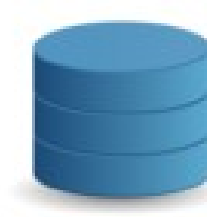
Le contratan para hacer una BD que permita apoyar la gestión de un sistema de ventas. La empresa necesita llevar un control de **proveedores, clientes, productos y ventas**. Un proveedor tiene un RUT, nombre, dirección, teléfono y página web. Un cliente también tiene RUT, nombre, dirección, pero puede tener varios teléfonos de contacto.

La dirección se entiende por calle, número, comuna y ciudad. Un producto tiene un id único, nombre, precio actual, stock y nombre del proveedor. Además se organizan en categorías, y cada producto va sólo en una **categoría**. Una **categoría** tiene id, nombre y descripción. Por razones de contabilidad, se debe registrar la información de cada venta con un id, fecha, cliente, descuento y monto final. Además se debe guardar el precio al momento de la venta, la cantidad vendida y el monto total por el producto.



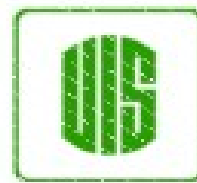
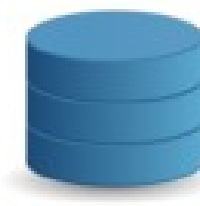
- Modelo ER Elementos>>Ejemplo





Diseño lógico de bases de datos utilizando creación de modelos de relaciones.

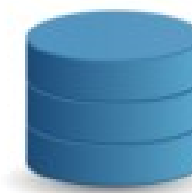




Diseño Lógico de Database.

Antes de implementar una base de datos, se diseña para que cumpla todos los requisitos. Esta primera tarea de diseñar se denomina diseño lógico

- **Creación de modelos de datos**
- **Entidades para diferentes tipos de relaciones**
- **Aplicación de reglas empresariales a relaciones**
- **Atributos para entidades**
- **Normalización para evitar redundancias**



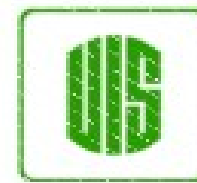
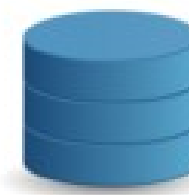
Diseño Lógico de Database.

Creación de modelos de datos:

La creación de modelos de datos lógicos es el proceso de documentación sobre los requisitos de información empresarial completos en un formato preciso y coherente.

Tenga en cuenta los siguientes estos pasos

- **Crean vistas de usuario críticas:** Esta etapa inicial del proceso de creación de modelos de datos es sumamente interactivo, se debe trabajar estrechamente con los usuarios reales.
- **Añaden reglas empresariales a vistas de usuario:** información detallada clave y las reglas empresariales más importantes. Que afectan a las operaciones de inserción, actualización y supresión realizadas en los datos.

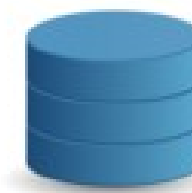


Diseño Lógico de Database.

Creación de modelos de datos:

- **Añaden detalles a las vistas de usuario y las valida:** Después de que los analistas han trabajado junto con los usuarios para definir las entidades y relaciones clave, añaden otros detalles descriptivos menos vitales. También asocian estos detalles descriptivos, llamados atributos, con las entidades.
- **Determinan reglas empresariales adicionales que afectan a los atributos:** Las reglas empresariales dirigidas a los datos son restricciones en valores de datos determinados. Estas restricciones deben cumplirse, independientemente de los requisitos de proceso particulares.

Ejemplo: cliente disponga de un número de teléfono o una dirección, o ambas cosas



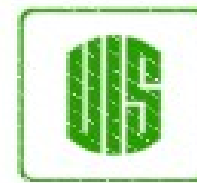
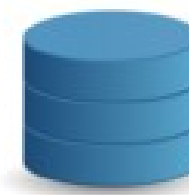
Diseño Lógico de Database.

Creación de modelos de datos:

- **Integran vistas de usuario:** los analistas combinan las distintas vistas de usuario que han creado para formar un modelo de datos lógico consolidado. Si ya existen otros modelos de datos en la organización, los analistas integran el nuevo modelo de datos en el existente.

Hacer que el modelo de datos sea flexible.

Ejem: empresa minorista trabaja en un único país y que dicha empresa planea añadir una ampliación a otros países



Diseño Lógico de Database.

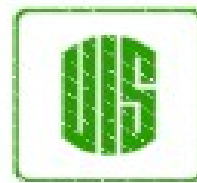
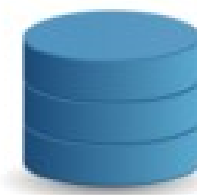
- **Aplicación de reglas empresariales a relaciones:**

Tanto si una relación determinada es de uno con uno, de uno con varios, de varios con uno o de varios con varios, es necesario que las relaciones tengan sentido empresarialmente.

- **Atributos para entidades :**

Cuando se definen atributos para las entidades, normalmente se trabaja con el administrador de datos para decidir los nombres, los tipos de datos y los valores adecuados para los atributo

- Convenios de denominación para atributos,
- Tipos de datos para atributos
- Valores para atributos de clave



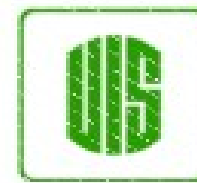
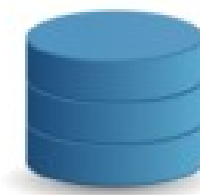
Diseño Lógico de Database.

- **Normalización para evitar redundancias:**

Después de definir las entidades y decidir los atributos para las entidades, se normalizan las entidades para evitar redundancias.

Una entidad está normalizada si cumple un conjunto de restricciones para una forma normal determinada, que se describe en esta información. Las entidades pueden tener las formas normales **primera, segunda, tercera y cuarta**, cada una de las cuales tiene unas determinadas reglas asociadas. En algunos casos, el usuario sigue estas reglas y, en otros casos, no las sigue.

Las reglas para la forma normal son acumulativas.



Diseño Lógico de Database.

- **Normalización para evitar redundancias:**

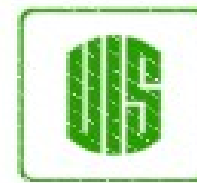
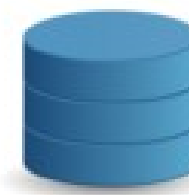
Después de definir las entidades y decidir los atributos para las entidades, se normalizan las entidades para evitar redundancias.

Una entidad está normalizada si cumple un conjunto de restricciones para una forma normal determinada, que se describe en esta información. Las entidades pueden tener las formas normales:

- **primera, segunda, tercera y cuarta**

cada una de las cuales tiene unas determinadas reglas asociadas. En algunos casos, el usuario sigue estas reglas y, en otros casos, no las sigue.

Las reglas para la forma normal son acumulativas.



Diseño Lógico de Database.

Normalización para evitar redundancias:

- **Primera forma normal**

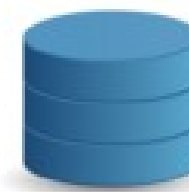
Una entidad relacional satisface el requisito de la primera forma normal si cada instancia de la entidad contiene únicamente un valor, pero varios atributos repetitivos.

Ejem.

Entidad Persona.

- Atributos fecha_nacimiento.
 - * edad.
 - * fecha_mayoria_edad

* estas dos sobran se puede calcular con la primera



Diseño Lógico de Database.

Normalización para evitar redundancias:

- **Segunda forma normal**

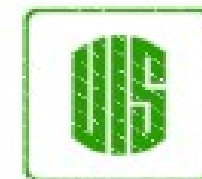
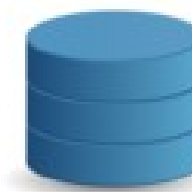
Una entidad cumple la segunda forma normal si cada atributo que no está en la clave primaria proporciona un hecho que depende de la clave completa. Se produce una violación de la segunda forma normal cuando un atributo de clave no primaria es un hecho sobre un subconjunto de una clave compuesta.

Ejem.

Habilidades de los empleos		
Empleo	Habilidad	Lugar de Trabajo
Hugo	Xbox	Pato 33 av 44
Paco	PS4	Pato 33 av 44
Luis	Xbox	Cll 33 cra 45
Hugo	PS4	Pato 33 av 44
Paco	Wii	Pato 33 av 44

Empleado	
Empleado	Lugar de Trabajo
Hugo	Pato 33 av 44
Paco	Pato 33 av 44
Luis	Cll 33 cra 45
Hugo	Pato 33 av 44
Paco	Pato 33 av 44

Habilidades de los Empleados	
Empleado	Habilidad
Hugo	Xbox
Paco	PS4
Luis	Xbox
Hugo	PS4
Paco	Wii



Diseño Lógico de Database.

Normalización para evitar redundancias:

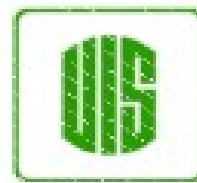
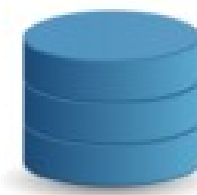
- **Tercera forma normal**
 - Ninguna columna puede depender de una columna que no tenga una clave.
 - No puede haber datos derivados.

ejem.

Codigo	nombre	lugar	Direccion
C034	Billie Eilish where do we go	Movistar Arena	Dg. 61c #26-36
C055	Banda Afrocacha	Teatro Colón	Calle 10 # 5 -32
C066	FreeDom 2020	El Teatro Jorge Eliécer Gaitán	Cra. 7 #22-47
C014	Interpol & Franz Ferdinand	Movistar Arena	Dg. 61c #26-36

Codigo	nombre
C034	Billie Eilish where do we go
C055	Banda Afrocacha
C066	FreeDom 2020
C014	Interpol & Franz Ferdinand

lugar	Direccion
Movistar Arena	Dg. 61c #26-36
Teatro Colón	Calle 10 # 5 -32
El Teatro Jorge Eliécer Gaitán	Cra. 7 #22-47
Movistar Arena	Dg. 61c #26-36



Diseño Lógico de Database.

Normalización para evitar redundancias:

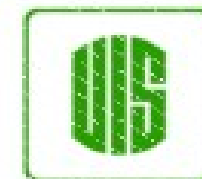
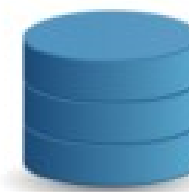
- **Cuarta forma normal**

Al igual que en las otras formas, la cuarta forma normal tiene como requisito el cumplir con su predecesora; si se cumple dicha condición, entonces se tienen que

eliminar todas las relaciones muchos a muchos

del modelo de datos para cumplir con la cuarta forma.

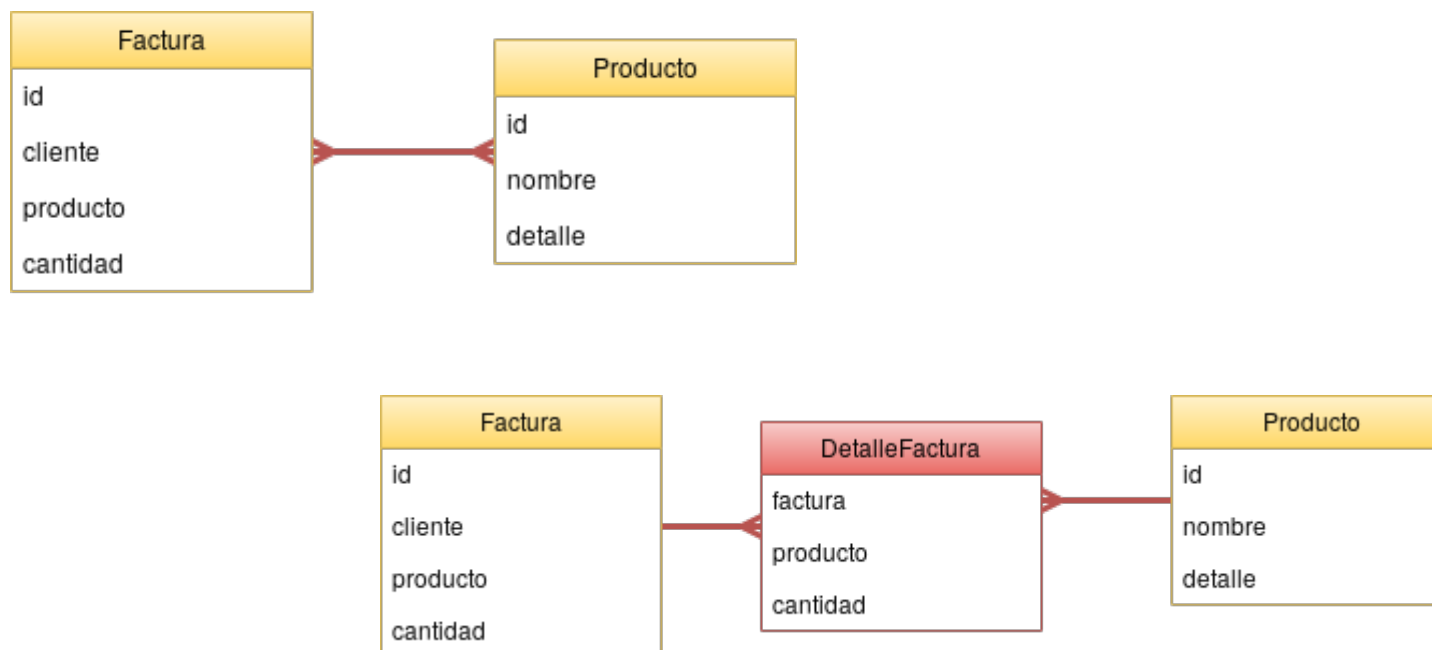
transforma las múltiples dependencias valoradas en dependencias funcionales esto implica que un valor de varios valores y no depende de la clave principal

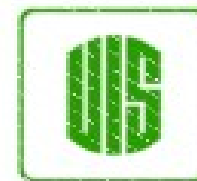
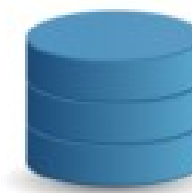


Diseño Lógico de Database.

Normalización para evitar redundancias:

- Cuarta forma normal





Diseño Lógico de Database.

Normalización para evitar redundancias:

- **Quinta forma normal**

La tabla está en 4FN

No existen relaciones de dependencias no triviales que no siguen los criterios de las claves. Una tabla que se encuentra en la 4FN se dice que está en la 5FN si, y sólo si, **cada relación de dependencia se encuentra definida por las claves candidatas**

se hace cuando una tabla tiene una gran cantidad de atributos y se hace inmanejable o cuando se tiene mucha información redundante con pocos atributos