

INGENIERIA REVERSA

SILVIO ENRIQUE BASANTE - HEBERT CARRILLO GUZMAN
ANDRES VARGAS CABAS

Estudiantes de 8o. Semestre de Ingeniería de Sistemas ICESI

INTRODUCCION

En muchas organizaciones hay un gran número de Bases de Datos que se han desarrollado en muchos años. Una gran dificultad con estas Bases de Datos es que frecuentemente el significado de los datos se ha perdido, además de que no se conoce exactamente cuáles son los datos y las relaciones entre ellos. Esta falta de entendimiento dificulta la utilización efectiva de los datos, dentro de una organización, y también reduce la probabilidad de que las actividades de mantenimiento puedan ser realizadas correctamente. Tal entendimiento puede ser alcanzado solamente por el aumento del nivel de abstracción sobre el de la Base de Datos misma, y representándolo como un modelo conceptual.

Ya que los modelos de Entidad Relación son los modelos conceptuales más ampliamente usados, el objetivo de esta investigación es presentar una metodología que extraiga un modelo Entidad Relación extendido de una Base de Datos Relacional existente.

Para hacerlo, los conceptos de Ingeniería del Software reversa, son aplicados. La metodología analiza no solamente el esquema de datos, sino también las instancias de los datos, las cuales contienen información detallada acerca del dominio de la aplicación.

OBJETIVO

El objetivo general de esta investigación es desarrollar una metodología de Ingeniería Reversa de Bases de Datos, la cual puede obtener, en un nivel alto de automatización, un "buen" modelo de Entidad Relación que corresponda a las especificaciones de diseño de una Base de Datos Relacional existente. Como resultado, necesitamos un entendimiento de la relación entre las especificaciones de diseño de la Base de Datos reconocida en un Modelo de Entidad Relación en la implementación de la Base de Datos.

1. METODOLOGIA PARA EL DESARROLLO DE INGENIERIA REVERSA

La metodología divide la Ingeniería Reversa de Bases de Datos Relacionales en los siguientes tres pasos:

1. *Clasificación*: Clasificar tablas y atributos.

2. *Generación*: Generar dependencias de inclusión.

3. *Identificación*: Identificar estructuras de modelación para el modelo EER.

Existen cuatro aspectos principales para tener en cuenta sobre la metodología:

- Suposiciones.
- Información requerida para el proceso de extracción.
- El proceso de extracción.
- Verificación y validación del proceso de extracción.

El proceso de extracción de la Ingeniería Reversa de Base de Datos Relacionales recupera semánticas de dominio de una Base de Datos existente y representa tales resultados como un Modelo EER.

1.1. Suposiciones

Tres grandes suposiciones deben ser propuestas acerca de las características de la Base de Datos de entrada.

1.1.1. Tablas en tercera forma normal

Una investigación previa ha estudiado cómo inferir dependencias funcionales analizando las instancias de los datos en la Base de Datos existente, es decir, que aunque la metodología no tiene una regla para saber si las tablas están en tercera forma normal, haciendo consultas a la Base de Datos y verificando que no haya duplicidad de datos y que no haya información que es falsa, teniendo en cuenta la integridad referencial y la integridad de entidad, podremos obtener alguna seguridad de que la Base de Datos está en tercera forma normal. Esta suposición simplifica el proceso de extracción ya que cada tabla corresponderá a un tipo de Entidad o un tipo de Relación, más que corresponder a varios tipos de Entidad o a

una mezcla de tipos de Relación y Entidad.

1.1.2. Nombramiento consistente de atributos clave

Una convención de nombramiento consistente es aplicada sobre los atributos clave, es decir, si los atributos clave tienen el mismo dominio, entonces tendrán el mismo nombre a lo largo de todas las tablas. Esta convención debe ser seguida en el proceso del diseño de la Base de Datos; de lo contrario será muy difícil para los usuarios de la Base de Datos hacer consultas que involucren varias tablas.

Esto permite que:

- Cada tabla y atributo de la Base de Datos de Entrada sean clasificados en su propia categoría en el paso 1, y
- Posibles dependencias de inclusión basadas en llaves sean formulars en el paso 2.

1.1.3. Datos no erróneos en valores de atributos clave

La Base de Datos de entrada no contiene instancias de datos erróneos en sus atributos clave. Esto permite que toda posible dependencia de inclusión sea descubierta en el paso 2.

1.2. Información requerida para el proceso de extracción

El proceso de extracción debe tener los siguientes tipos de información, en este orden:

1.2.1. Esquema de datos

Inicialmente, la información acerca del esquema de datos debe estar disponible. Esta incluye nombres de tablas, nombres de atributos y llaves primarias. En general, la información acerca de los nombres de atributos y tablas puede ser obtenida directamente consultando el DBMS. Si el DBMS no contiene información acerca de las llaves primarias,

el usuario debe especificar la llave primaria para cada tabla.

La información acerca de las propiedades de los atributos, tales como atributos únicos, atributos de valores únicos y no nulos, puede ser aplicada para reducir el número de atributos posibles para la llave primaria de cada tabla.

En nuestra metodología, la información acerca de llaves candidatas es requerida sólo cuando:

- Hay ambigüedades en la clasificación de tablas, y
- El usuario especifica dependencias de inclusión entre atributos no claves.

1.2.2. Instancias de datos

El proceso de extracción analiza las instancias de datos para:

- Verificar las dependencias de inclusión propuestas, e
- Identificar llaves candidatas.

1.2.3. Dependencias de inclusión

Las dependencias de inclusión son necesarias para identificar:

1. Relaciones de inclusión
2. Tipos de entidad padre para entidades débiles, y
3. Tipos de entidad participantes para relaciones.

La generación de dependencias de inclusión basadas en claves es un proceso automático. Además, el proceso de extracción permite al usuario especificar las dependencias de inclusión entre atributos no claves.

1.2.4. Semánticas de dominio obtenidas del usuario

Debido a la limitada expresión semántica de las corrientes DBMS, la Ingeniería Reversa de Bases de Datos no puede ser un proceso totalmente automático. El usuario involucrado es muy necesario, sobre todo cuando las

semánticas de dominio no pueden ser inferidas o haya ambigüedades que no puedan ser resueltas mecánicamente.

1.2.5. Heurísticas

Ciertos tipos de heurísticas son usadas para:

1. Proponer atributos de llave primaria para cada tabla.
2. Formular posibles dependencias de inclusión basadas en claves, y
3. Especificar relaciones de cardinalidad por defecto para tipos de relaciones binarias.

1.2.6. Reglas de extracción

Las reglas de extracción son usadas para:

- Clasificar tablas y atributos (Reglas de Clasificación),
- Generar dependencias de Inclusión (Reglas de Inferencia), y
- Convertir atributos y tablas clasificadas en estructuras modeladas correspondientes al modelo EER (Reglas de Identificación y Asignación).

1.3. Proceso de extracción

El proceso de extracción consta de los siguientes tres pasos:

1. Clasificación de tablas y atributos,
2. Generación de dependencias de inclusión.
3. Identificación de las entidades y de los tipos de relaciones.

1.3.1. Clasificación de tablas y atributos

1.3.1.1. Tabla de entidad fuerte

Una tabla de entidad fuerte es aquella cuya llave primaria no contiene propiamente una llave de cualquier otra tabla. En nuestro ejemplo, EMPLEADO, DIRECTOR y DEPARTAMENTO son ejemplos de tablas de entidad fuerte. TRABAJA_PARA no es una tabla de entidad fuerte porque su llave primaria, (SSN, DEPTNO), contiene las llaves

primarias de EMPLEADO o DIRECTOR y DEPARTAMENTO.

1.3.1.2. Tabla de entidad débil

Una tabla de entidad débil es una tabla en la que se encuentran los siguientes requerimientos:

1. Un subgrupo propio de su llave primaria, llamado K1, contiene las llaves de otras tablas de entidad (fuerte y/o débil).
2. Los atributos sobrantes de la llave primaria, llamados K2, no son llaves de otras tablas.
3. Esta tabla representa un tipo de entidad, cuyo identificador debe contener atributos identificadores de otros tipos de entidad. Este requerimiento debe ser confirmado por un usuario experto.

En nuestro ejemplo, consideremos PROYECTO. Su llave primaria contiene la llave primaria DEPTNO de la entidad DEPARTAMENTO, la cual es una tabla de entidad fuerte. Si PROJNAME no es una llave de ninguna otra tabla, y PROYECTO representa un tipo de entidad, entonces PROYECTO es una tabla de entidad débil.

1.3.1.3. Tabla de relación regular

Una tabla de relación regular es aquella cuya llave primaria está formada totalmente por la concatenación de llaves de tablas de entidad (fuerte y/o débil). En nuestro ejemplo, la tabla PUEDE_PRODUCIR, está formada totalmente por la concatenación de las llaves primarias de DEPARTAMENTO y PRODUCTO o PRECIO.

1.3.1.4. Tabla de relación específica

Una tabla de relación específica es una tabla cuya llave primaria está parcialmente formada por la concatenación de llaves de tabla de entidad (fuerte y/o débil), y no puede ser tratada como una tabla de entidad débil. En nuestro ejemplo, consideremos ENVIO, PAQUETE #

no es una llave de otra tabla entidad. Si ENVIO no puede satisfacer el tercer requerimiento para las tablas de entidad débil, entonces ENVIO es una tabla de relación específica.

1.3.1.5. Atributo Clave Primario (PKA)

Los atributos clave primario son:

1. Atributos de llaves primarias de tablas entidades fuertes,
2. Atributos de llaves primarias de tablas de entidad débil, los cuales aparecen como llaves para algunas tablas de entidad, y
3. Atributos de llaves primarias de tablas de relación, las cuales son también llaves de otras tablas.

1.3.1.6. Atributo clave pendiente (DKA)

Los atributos de la llave primaria de una tabla de entidad débil, los cuales no aparecen como atributos claves de cualquier otra tabla, son atributos claves pendientes.

1.3.1.7. Atributo clave general (GKA)

Los atributos de la llave primaria de una tabla de relación específica, que no aparecen como atributos claves de cualquier otra tabla, son atributos claves generales.

1.3.1.8. Atributo clave foráneo (FKA)

Si un subgrupo de atributos que no son llave primaria de una tabla dada aparecen como una llave de otra tabla de entidad (fuerte y/o débil) son atributos claves foráneos.

1.3.1.9. Atributo no clave (NKA)

Son aquellos atributos sobrantes que no son atributos claves.

En nuestro ejemplo, DEPTNO en DEPARTAMENTO y (SSN, DEPTNO) EN TRABAJA-PARA son atributos claves primarios. PROJNAME es clasificado como un atributo clave pendiente porque PROYECTO es una tabla de entidad débil. El atributo DEPTNO en

DIRECTOR es un atributo clave foráneo porque aparece como la llave primaria de DEPARTAMENTO. Los atributos DEPT-NAME y LOCALIZACION en DEPARTAMENTO son atributos no claves.

La información acerca de las llaves primarias es usada en primera medida para clasificar tanto las tablas de entidad fuerte como esas tablas de relación regular que contienen sólo llaves primarias de tablas de entidad fuerte. Luego, las tablas restantes, cuyas llaves primarias contienen atributos que no aparecen como llave primaria de otra tabla de entidad fuerte, son clasificadas.

Suponga que los atributos de llave primaria, que no aparecen como atributos de llave primaria en otra tabla, aparecen como atributos no primarios en tablas de entidad fuerte y además son verificadas como llaves candidatas por análisis de las instancias de los datos. Por lo tanto esta tabla es clasificada como una tabla de relación regular.

En nuestro ejemplo, PROYECTO: (PROJNAME, DEPTNO, PRESUPUESTO). Si PROJNAME es una llave candidata para una tabla de entidad fuerte, entonces PROYECTO es clasificada como una tabla de relación regular. Esto significa que una tabla de entidad fuerte tiene a PROJNAME como un atri-

buto no clave que contiene valores nulos y únicos de instancias de datos. La información acerca de las llaves candidatas será usada en la formulación de dependencias de inclusión.

Si un subgrupo propio de atributos de llaves primarias de una tabla no aparece como una llave en alguna otra tabla, entonces es imposible clasificar esta relación como una tabla de entidad débil o una tabla de relación específica aun por análisis de las instancias de los datos. Para clasificar tal tabla, el proceso de extracción debe referirse a las semánticas del dominio las cuales son capturadas en la fase de diseño pero no representadas en el esquema de datos.

En nuestro ejemplo, ENVIO (PAQUETE #, ORDID, FECHA-ENVIO, CARRIER-ID). Suponga que PAQUETE # no aparece como una llave de otra tabla de entidad fuerte (es decir que ENVIO satisface los requerimientos 1 y 2 de la definición de tabla de entidad débil). Entonces el usuario debe confirmar si ENVIO es una tabla de entidad débil o una tabla de relación específica. ENVIO es clasificada como una tabla de entidad débil, si también satisface el tercer requerimiento para una tabla de entidad débil; de lo contrario, es una tabla de relación específica.

Tabla	Tipo	PKA	DKA	GKA	FKA	NKA
Persona	Fuerte	{SSN}				{Nombre, Dirección}
Empleado	Fuerte	{SSN}				{Nombre, salario, fecha_inicio}
Director	Fuerte	{SSN}			{dept}	
Cliente	Fuerte	{CUSTID}			{ssn}	{nombre, crédito}
Departamento	Fuerte	{DEPT}				{Nom_depto, locación}
Producto	Fuerte	{PRODID}				{descripción}
Precio	Fuerte	{PRODID}				{minprecio, maxprecio}
Orden	Fuerte	{ORDID}			{prodid}	{fecha_orden, custid}
Carrier	Fuerte	{CARID}				{nombre, dirección}
Proyecto	Débil	{DEPT}	{NOM}			{budget}
Trabaja-para	Reg	{SSN,Dept}				{fecha_inicio}
Puede-Producir	Reg	{DEPT, PRODID}				{costo_unitario}
Envío	Especi	{ORDID}		{PAQ#}	{carid}	{fecha_envío}

Clasificación de relaciones y atributos

1.3.2. Generación de dependencias de inclusión

Las dependencias de inclusión contienen la mayoría de la información para la identificación de los tipos de relación. El proceso de extracción detecta las posibles dependencias de inclusión refiriéndose a los atributos y tablas clasificadas, y rechazando las inválidas, analizando las instancias de los datos.

Utilizaremos unas heurísticas y reglas de inferencia para esta generación.

1.3.2.1. Heurísticas para proponer posibles dependencias de inclusión

Como el primer paso es formular posibles dependencias de inclusión con el

Persona {SSN, Nombre, Dirección}	(fuerte)
Empleado {SSN, Nombre, Salario, Fecha_Contrato}	(fuerte)
Director {SSN, Rango, Fecha_Promoción, Deptno}	(fuerte)
Cliente {SSN, ID_Cliente, Nombre, Crédito}	(fuerte)
Producto {Prodid, Descripción}	(fuerte)
Precio {Prodid, Preciomin, Preciomáx}	(fuerte)

Las siguientes parejas de atributos claves son considerados:

- (Persona. {SSN}, empleado. {SSN})
- (Persona. {SSN}, Director. {SSN})
- (Persona. {SSN}, Cliente. {SSN})
- (Empleado. {SSN}, Director. {SSN})
- (Empleado. {SSN}, Cliente. {SSN})
- (Director. {SSN}, Cliente. {SSN})
- (Producto. {Prodid}, Precio. {Prodid})

En nuestro ejemplo:

Director {SSN, Rango, Fecha_Promoción, Deptno}	(fuerte)
Departamento {Deptno, Nombre_Dept, Localización}	(fuerte)
Envío {Paquete #, Ordí, Fecha, Envío, Carrier_Id}	(fuerte)
Carrier {Carrier_Id, Nombre, Dirección}	(fuerte)

fin de evitar la formulación de muchas dependencias inapropiadas, el proceso de extracción utiliza heurísticas que solamente formulan dependencias de inclusión entre atributos claves de tablas. Las heurísticas son:

⇒ SI: dos tablas de entidad fuerte, A y B, tienen la misma llave, X:

Entonces: puede haber una dependencia de inclusión entre A, X y B.X; es decir, ó A.X << B.X. ó B.X. << A.X.

Justificación: la existencia de relaciones subtipo/supertipo entre tipos de entidad, surgen en tales dependencias de inclusión. Además, estas dependencias de inclusión son usadas para identificar relaciones de inclusión.

Considere las siguientes tablas:

⇒ SI: la llave X, de una tabla de entidad (fuerte/débil), A, aparece como llave foránea, X, de otra tabla (entidad o relación), B;

Entonces: puede haber una dependencia de inclusión entre B. X y A.X, denotada por B.X << A.X.

Justificación: los casos donde las relaciones binarias y relaciones entre tipos de relación y entidad son representadas por llaves foráneas, pueden ser dependencias de inclusión.

Las siguientes parejas de atributos claves son consideradas:

(Director. {Deptno}, Departamento. {Deptno})

(Envío. {Carrier_Id}, Carrier. {Carrier_Id})

⇒ SI: los atributos de la llave primaria, X, de una tabla de relación (regular o específica) o una tabla de entidad débil, S, aparece como una llave de una tabla de entidad, A;

Entonces: podrá haber una dependencia de inclusión entre S.X y A.X denotada por S.X << A.X.

Justificación: la presencia de relaciones representadas por tablas de relación o entidades débiles resultarán en este tipo de dependencia de inclusión. Estas dependencias son usadas para determinar los tipos de entidad padres para entidades débiles y tipos de entidades participantes para relaciones.

En nuestro ejemplo:

Trabaja-Para: {SSN, Deptno, Fecha_Inicio} (regular)

Proyecto: {Projname, Deptno, Presupuesto} (regular)

Empleado: {SSN, Nombre, Salario, Fecha_Contrato} (regular)

Director: {SSN, Rango, Fecha_Promoción, Deptno} (regular)

Departamento: {Deptno, Nombre_Dept, Localización} (regular)

Las siguientes parejas de atributos claves son consideradas:

(Trabaja-Para. {SSN}, Empleado. {SSN})

(Trabaja-Para. {SSN}, Director. {SSN})

(Trabaja-Para. {Deptno}, Departamento. {Deptno})

(Proyecto. {Deptno}, Departamento. {Deptno})

Las anteriores heurísticas no son suficientes para detectar todas las posibles dependencias de inclusión basadas en llaves. Suponga que una llave candidata de una tabla de entidad fuerte, la cual no es identificada en la clasificación de tablas, aparece como un atributo no llave en otra tabla, entonces una dependencia de inclusión basada en llave puede existir.

En nuestro ejemplo, el atributo Custid en Cliente es una llave candidata y también aparece como un atributo no llave en Orden. El proceso de extracción, además, permite al usuario especificar dependencias de inclusión adicionales entre atributos no claves. Por ejemplo, el usuario especifica una dependencia de inclusión, Orden. {Custid} << Cliente. {Custid}. Luego, el atributo Custid en Cliente debe ser verificado como llave candidata por el análisis de las instancias de datos.

1.3.2.2. Rechazo de dependencias de inclusión inválidas

Cada una de las dependencias de inclusión propuestas está sujeta a más análisis. Dos reglas son usadas para rechazar dependencias de inclusión inválidas. Sean (A.X, B.X) un par de atributos. Valor(A.X) denota el conjunto de valores de A.X, y Valor(B.X) denota el conjunto de valores de B.X.

⇒ SI: Valor(B.X) es diferente de vacío y Valor(B.X) está incluido o es igual a Valor(A.X);

Entonces: B.X << A.X no puede ser rechazada.

⇒ SI: Valor(A.X) es diferente de vacío y Valor(A.X) está incluida o es igual a Valor(B.X);

Entonces: A.X << B.X no puede ser rechazada.

Justificación: estas reglas están basadas en la definición de dependencias de inclusión.

Para cada dependencia de inclusión propuesta, el proceso de extracción debe comparar los grupos de valores que aparecen en A.X. y B.X. Por ejemplo, las tablas Empleado y Cliente podrían tener la misma llave primaria, SSN. Es imposible, sin embargo, que haya un subgrupo de relación entre Empleado. {SSN} y Cliente. {SSN}, por lo tanto, esta dependencia es rechazada.

Consideremos la dependencia de inclusión propuesta entre Director. {SSN} y Empleado. {SSN}. Si el grupo de valores de Director. {SSN} es un subgrupo propio del grupo de valores de Empleado. {SSN}, entonces Director. {SSN} << Empleado. {SSN} no puede ser rechazada.

Si los atributos de llaves foráneas de una tabla no aparecen en alguna de las dependencias de inclusión válida, entonces ellas deben ser reclasificadas como atributos no claves.

1.3.2.3. Remover dependencias de inclusión redundantes

Las dependencias de inclusión redundantes pueden ser detectadas por reglas de inferencia de dependencias de inclusión (proyección y transitividad). Esto es:

⇒ Si: A.X << B.X y B.Y << C.Y se mantienen, y Y es subgrupo de X;

Entonces: la dependencia de inclusión A.Y << C.Y es redundante.

Justificación: las reglas de inferencia para dependencias de inclusión.

Hay dos razones para la eliminación de dependencias de inclusión redundantes. Primero, una dependencia de inclusión redundante puede manejar a una relación redundante ES-UN. En nuestro ejemplo tenemos:

Director.{SSN}<< Empleado.{SSN}
Empleado.{SSN}<< Persona.{SSN}
Director.{SSN}<< Persona. {SSN}.

La última dependencia de inclusión es redundante debido a que puede ser inferida de las dos primeras. Si no fuera eliminada, entonces una relación ES-UN de Director a Persona sería identificada. Esta relación es obviamente redundante porque existen unas relaciones ES-UN de Director a Empleado y de Empleado a Persona.

Segundo, una dependencia de inclusión redundante puede identificar el tipo de entidad participante requerido para un tipo de relación. En nuestro ejemplo tenemos:

Trabaja-Para.{Deptno}<< Departamento. {Deptno}

Trabaja-Para.{SSN}<< Empleado. {SSN}

Empleado.{SSN}<< Persona. {SSN}

Trabaja-Para.{SSN}<< Persona. {SSN}.

La última dependencia de inclusión es redundante. Si es usada para inferir tipos de entidad participante en un tipo de relación, trabaja-para, identificada por Trabaja-Para, entonces el tipo de entidad, persona, identificada por Persona se convierte en un tipo de entidad participante. Sin embargo, persona no debería ser inferida como el tipo de entidad participante de trabaja-para basada en la relación ES-UN de empleado a persona y la relación binaria entre empleado y departamento.

1.3.3. Identificación de tipos de entidad

Los tipos de entidad deben ser identificados primero para que los tipos de relación entre ellos puedan ser identificados de ahí en adelante. La existencia de una tabla de entidad indica la presencia de un tipo de entidad. Tablas de entidad fuerte identifican tipos de entidad fuerte, mientras que tablas de entidad débil, identifican tipos de entidad débil y las correspondientes relaciones identificadoras. El nombre de la tabla de entidad será el tipo de entidad correspondiente.

El tipo de entidad fuerte se identifica con la correspondiente llave primaria de la tabla de entidad fuerte de la cual surgió. El tipo de entidad débil se identifica con los correspondientes atributos claves derivados (DKA) de la tabla de entidad débil de la cual surgió.

En nuestro ejemplo,

Empleado:{SSN, Nombre, Salario, Fecha-Contrato} (fuerte)



Para el tipo de entidad débil se debe identificar la relación identificadora entre ésta y su(s) padre(s) identificador(es). Los tipos de entidad padre son determinados refiriéndose a las dependencias de inclusión. Usamos la siguiente regla para identificar estas relaciones de los tipos de entidad débil con su(s) padre(s):

- ⇒ Si: (1) un tipo de entidad débil es identificada por una tabla de entidad débil, W,
- (2) los atributos de la llave primaria, X, de W aparece como llave de una tabla de entidad (fuerte/débil), A, y
- (3) hay una dependencia de inclusión, W.X << A.X;

Entonces: el tipo de entidad identificado por A es el tipo de entidad padre del tipo de entidad débil identificado por W.

⇒ Si: más de una tabla de entidad satisface las anteriores condiciones;

Entonces: el usuario debe determinar la entidad padre exacta.

Justificación: las anteriores tres condiciones son necesarias para el tipo de

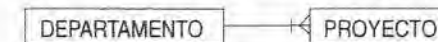
entidad identificado por A para ser un tipo de entidad padre del tipo de entidad débil identificado por W.

La cardinalidad máxima usual para una relación identificadora binaria es 1:N. En nuestro ejemplo, consideremos a Proyecto. Un tipo de entidad débil,

Proyecto, es identificado con su propio atributo clave derivado (DKA),

Projname, como llave. Proyecto tiene una relación identificadora con su tipo de entidad padre, Departamento, la cual es determinada por la dependencia de inclusión.

Proyecto.{Deptno}<<Departamento.{Deptno}



1.3.4. Identificación de tipos de relación

El proceso de extracción identifica cinco tipos de relaciones. Estas son:

1.3.4.1. Relaciones de inclusión

Se usan las dependencias de inclusión para identificar relaciones de inclusión.

- ⇒ Si: (1) Dos tablas de entidad fuerte, A y B, tienen la misma llave, X, y
- (2) Hay una dependencia de inclusión, A.X << B.X; entre sus llaves;

Entonces: identificar una relación ES-UN de cardinalidad mínima (1,0) entre los tipos de entidad de las correspondientes tablas de entidad.

Justificación: las relaciones binarias tienen al menos uno de los valores min/max (1,1) que son las únicas estructuras EER que manejan tales estructuras relacionales. Así una relación ES-UN es la interpretación más apropiada de las estructuras relacionales que satisfacen las anteriores condiciones.

En nuestro ejemplo, consideremos a Empleado y Director, dos tablas de enti-

dad fuerte, y la dependencia de inclusión.

Director.{SSN}<<Empleado.{SSN}. La dependencia de inclusión sugiere que existe una relación ES-UN de Director a Empleado. La cardinalidad máxima para una relación ES-UN es siempre 1:1.



⇒ SI:(1) dos tablas de entidad fuerte, A y B, tienen la misma llave, X, y

(2) hay dos dependencias de inclusión, $A.X \ll B.X$ y $B.X \ll A.X$, entre sus llaves;

Entonces: identificar una relación de inclusión con cardinalidad mínima (1:1) entre los tipos de entidad identificadas por A y B respectivamente

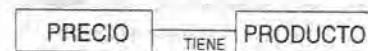
Justificación: A y B tienen no sólo la misma llave sino también el mismo grupo de valores en sus llaves, puede ser mejor interpretada con la existencia de una relación de inclusión. La relación de inclusión es representada como un tipo de relación binaria en el modelo EER con un nombre propio.

En nuestro ejemplo, consideremos Producto y Precio y las dos dependencias de inclusión:

Producto. {Prodid}<<Precio. {Prodid}

Precio. {Prodid}<<Producto. {Prodid}

Producto y Precio son las tablas de entidad fuerte. Ellas tienen no sólo la misma llave primaria sino también el mismo grupo de instancias de datos para sus llaves primarias.



1.3.4.2. Relaciones binarias

Las llaves foráneas de tablas de entidad y las dependencias de inclusión especificadas por el usuario son usadas

para identificar tipos de relaciones binarias. Primero, una llave foránea en una tabla de entidad indica que un tipo de relación binaria existe.

La regla de identificación es:

⇒ SI: (1) una llave foránea, X, de una tabla de entidad (fuerte/débil), B, aparece como una llave, X, de otra tabla de entidad (fuerte/débil), A,

(2) hay una dependencia de inclusión entre ellas, $B.X \ll A.X$;

Entonces: identificar un tipo de relación binaria, la cual tiene dos tipos de entidades participantes. Una es identificada como la tabla que contiene la llave foránea.

⇒ SI: más de una tabla de entidad, A, satisface las anteriores condiciones.

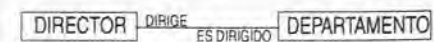
Entonces: el usuario debe determinar el tipo de entidad participante exacto.

Justificación: las llaves foráneas en tablas de entidad son usadas para representar relaciones binarias entre tipos de entidad. Así, cada llave foránea de una tabla de entidad es convertida en un tipo de relación binaria.

Ya que el nombre de tales relaciones no es normalmente almacenado en el DBMS, el usuario debe especificarlo. La cardinalidad máxima para un tipo de relación binaria identificada por una llave foránea es ó 1:1 ó 1:N. Si la llave foránea contiene valores únicos, entonces la cardinalidad máxima es 1:1.

En nuestro ejemplo consideremos los tipos de entidad Departamento y Director, y una dependencia de inclusión,

Director.{Deptno}<<Departamento.{Deptno}. El proceso de extracción identifica una relación binaria entre los tipos de entidad Director y Departamento.



Si la llave foránea es también una llave candidata, entonces la tabla binaria podría ser una relación ES-UN con cardinalidad mínima (1:0).

En nuestro ejemplo tenemos:

Persona:

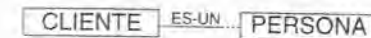
{SSN, Nombre, Dirección} (fuerte)

Cliente:

{Custid, SSN, Nombre, Crédito} (fuerte)

Cliente.{SSN}<<Persona.{SSN}

En este caso, la tabla de entidad fuerte, Cliente, tiene su llave primaria, CUSTID, en vez de ssn, la cual es una llave foránea. Por lo tanto, hay una relación ES-UN de Cliente hacia Persona con cardinalidad mínima (1:0).



Segundo, una dependencia de inclusión válida entre atributos no claves (uno de los cuales es una clave candidata), especificada por el usuario, indica que un tipo de relación binaria debe ser identificada. La regla de identificación es:

⇒ SI: hay una dependencia de inclusión entre atributos no claves, $A.X \ll B.Y$;

Entonces: identificar un tipo de relación binaria consistente de dos tipos de entidad participantes identificada por las tablas A y B.

1.3.4.3. Relaciones identificadas por tablas de relaciones

Una tabla de relación indica la presencia de una asociación entre tipos de entidad. Para cada tabla de relación, el proceso de extracción identifica un tipo de relación n-aria entre los tipos de entidad participantes cuyas llaves forman la llave primaria de esta tabla. El nombre de la tabla de relación es asignado para ser el nombre del tipo de relación

correspondiente. Como mencionamos anteriormente, hay dos tipos de tablas de relación: regular y específica. La manera en la cual son convertidas en tipos de relación es diferente. Para cada tabla de relación regular, el proceso de extracción identifica un tipo de relación entre tipos de entidad participantes que ya han sido identificados por el proceso de extracción. Para cada tabla de relación han sido identificados por el proceso de extracción. Para cada tabla de relación específica, sin embargo, uno o más tipos de entidad deben ser identificados primero. Luego, un tipo de relación es identificado entre todos los tipos de entidad participantes.

Los tipos de entidad participantes son determinados al examinar las llaves primarias y las dependencias de inclusión. El grado de un tipo de relación es determinado por el número de tipos de entidades participantes. El proceso de extracción asigna una cardinalidad máxima de N: M para un tipo de relación binaria identificada por la tabla de relación.

Tablas de relación regular. Una tabla de relación regular identifica un tipo de relación n-aria. La regla de identificación es:

⇒ SI: hay una tabla de relación regular;

Entonces: identificar un tipo de relación n-aria entre los tipos de entidad cuyas llaves forman la llave primaria de la tabla.

Justificación: una tabla de relación regular puede ser convertida en tipo de relación n-aria o un tipo de relación entre un tipo de entidad y un tipo de relación si su grado es mayor que 2.

Los tipos de entidad participantes para un tipo de relación identificada de una tabla de relación regular son determinados por la siguiente regla:

⇒ SI: (1) un subgrupo, X, de los atributos de la llave primaria

- **Rit_Usuario** (Cod_Usuario, Nombre_Usuario, Semes_Usua, Plan_Usuario, Clase_Usuario).
- **RIT_Reserva** (Cod_Usua, Fecha_Reserva, Hora_Inicial, Hora_Final, Cod_Usa, Cod_Equipo, Sala, Cod_Plan, Tipo_Reserva, Confirmación).

El paso siguiente sería indicar cuáles son los atributos que hacen parte de la clave primaria de cada tabla, al igual que los atributos que hacen parte de la clave foránea.

Las tablas quedarían en la siguiente forma:

Rit_Equipos

Cod_Equipo	Des_Equipo	Est_Equipo	Sala
Pk			Pk

Rit_Espec.

Cod_Usuario	Nombre_Usuario	Plan_Usuario
Pk		

Rit_Plan

Cod_Plan	Nombre_Plan
Pk	

Rit_Reserva

Cod_Usuario	F_Reserva	H_inicial	H_final	Cod_Usa	Cod_Equipo	Confirman.	T_reserva	Sala	Cod_Plan
Pk	Pk			Fk	Fk			Pk	Fk

Rit_Usuario

Cod_Usuario	Nombre_Usuario	Semes_Usuario	Plan_Usuario	Clase_Usuario
Pk				

Rit_Uso

Cod_Usa	Nombre_Usa
Pk	

A continuación realizaremos el primer paso del proceso de extracción, el cual consiste en la clasificación de las tablas y atributos.

Primeramente, analizaremos la tabla Rit_Equipos; como podemos observar, en su tabla aparecen dos llaves primarias: Cod_Equipo y Sala; estos atributos son propios de la tabla, por esta razón y teniendo en cuenta la definición de una entidad fuerte, podemos deducir que la Tabla Rit_Equipos es una Tabla de Entidad Fuerte.

Al realizar un análisis semejante a las tablas Rit_Plant, Rit_Usuarios, Rit_Usos y Rit_Espec, encontramos que cumplen exactamente con las mismas condicio-

nes de la tabla Rit_Equipos, es decir, que sus atributos de la llave primaria pertenecen únicamente a esa tabla específica. Por esta razón, las tablas anteriormente mencionadas son clasificadas como Tablas de Entidad Fuerte.

Para la tabla Rit_Reserva, tenemos que su clave primaria está conformada por Cod_Usuario, Fecha_Reserva y Sala; donde Cod_Usuario y Sala son llaves de otras tablas, mientras que Fecha_Reserva no pertenece a ninguna otra llave, clasificándola como una llave derivada para la tabla a la cual pertenece. Como podemos observar, esta tabla cumple con todos los requerimientos de una Tabla de Entidad Débil.

Clasificación de las Tablas y sus Atributos:

Tabla	Tipo	PKA	DKA	GKA	FKA	NKA
Rit_Equipos	Fuerte	[Cod_equipo, sala]				[Des_Equipo, Est_Equipo]
Rit_Espec	Fuerte	[Cod_Usuario]				[Nombre_Usuario, Plan_Usuario]
Rit_Plan	Fuerte	[Cod_Plan]				[Nombre_Plan]
Rit_Reserva	Débil	[Cod_Usuario, Sala]	[Fecha Reserva]		[Cod_uso, cod_plan, cod_Eq]	[Hora_Inicial, Hora_final, confirma, Tipo_Reserva]
Rit_Usos	Fuerte	[Cod_Usos]				[Nombre_Usa]
Rit_Usuarios	Fuerte	[Cod_Usuario]				[Nombre_usua, Semes_Usua, Plan_usua, Clase_U]

El paso siguiente del proceso de extracción es realizar la generación de dependencias de inclusión, siguiendo paso a paso las heurísticas descritas anteriormente para tal fin.

Las entidades Rit_Espec y Rit_Usuarios, fueron clasificadas como entidades fuertes y además poseen la

misma llave primaria Cod_usuario; al aplicarle la heurística 1 debemos crear una dependencia de inclusión entre Rit_Espec, Cod_Usuario y Rit_Usuario, Cod_Usuario, la cual es Rit_Espec. {Cod_Usuario}<<

Rit_Usuario. {Cod_Usuario} Rit_Usuario. {Cod_Usuario}<<

Rit_Espec. {Cod_Usuario}; que transformándola a una pareja de atributo clave sería:

(Rit_Espec. {Cod_Usuario}, Rit_Usuario. {Cod_Usuario})

La llave Cod_uso, es la llave primaria de Rit_Usos, la cual es una entidad fuerte, pero también aparece como una llave foránea de la entidad Rit_Reserva. Esta llave cumple con los requerimientos de la heurística 2, razón por la cual debemos crear una dependencia de inclusión entre Rit_Reserva. {Cod_uso} y Rit_Usos. {Cod_Usos}, denotada por Rit_Reserva. Cod_Usos}<<

Rit_Usos. {Cod_Usos}; que transformándola a una pareja de atributo clave sería:

(Rit_Reserva. {Cod_Usos}, Rit_Usos. {Cod_Usos}).

De la misma manera podemos aplicar esta heurística a las llaves Cod_Plan y Cod_Equipo, las cuales son llaves primarias de las entidades Rit_Plan y Rit_Equipos respectivamente. Estas llaves aparecen también como llaves foráneas en la tabla Rit_Reserva, por tal razón debemos incluir otras dos dependencias de inclusión, las cuales siguiendo los mismos procesos que en el caso anterior serían:

(Rit_Reserva. {Cod_Plan} y Rit_Plan. {Cod_Plan}),

(Rit_Reserva. {Cod-equipo} y Rit_Equipos. {Cod_Equipo}).

En el caso práctico se ha clasificado a Rit_Reserva como una entidad débil, la cual contiene en su llave primaria los atributos Cod_Usuario y Sala; donde Cod_Usuario aparece como llave en las entidades Rit_Usuario y Rit_Espec y, Sala aparece como llave en la entidad Rit_Equipos. Si se realiza un seguimiento a la heurística 3, se podrá observar que se requieren otras tres dependencias de inclusión entre estas entidades, que llevadas a parejas de atributo clave, serían de la siguiente manera:

(Rit_Reserva. {Cod_Usuario} y Rit_Usuario. {Cod_Usuario}),

(Rit_Reserva. {Cod_Usuario} y Rit_Espec. {Cod_Usuario}),

(Rit_Reserva. {Sala} y Rit_Equipos. {Sala}).

Después de haber generado todas las dependencias posibles con la aplicación de las heurísticas, se verificó que hubiese dependencias redundantes así como lo indica la metodología.

Ahora, siguiendo el desarrollo de la metodología, se entrará en la etapa de la identificación de los tipos de entidad. Para el caso en estudio se definieron únicamente dos tipos de entidades, las cuales fueron Fuertes y Débil. Por esta razón aparecen las siguientes entidades:

Fuertes:

RIT_EQUIPO
COD_EQUIPO
SALA
* DES_EQUIPO
* EST_EQUIPO

RIT_ESPEC
COD_USUARIO
* NOMBRE_USUA
* PLAN_USUARIO

RIT_USUARIO
COD_USUARIO
* NOMBRE_USUA
* SEMES_USUA
* PLAN_USUARIO
* CLASE_USUA

RIT_PLAN
COD_PLAN
* NOMBRE_PLAN

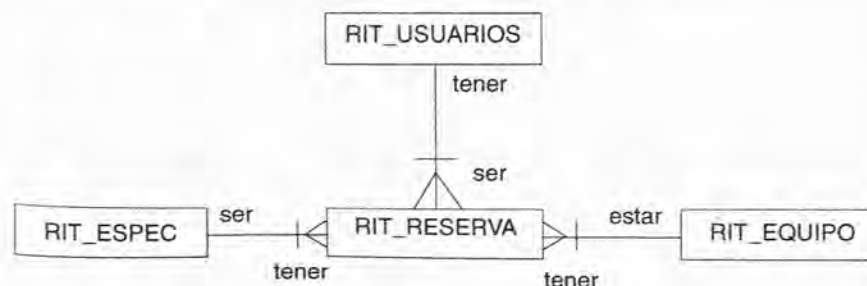
RIT_USOS
COD_USO
* NOMBRE_USO

Débil:

RIT_RESERVA
FECHA_RESERVA
* HORA_INICIAL
* HORA_FINAL
* CONFIRMACION
* TIPO_RESERVA

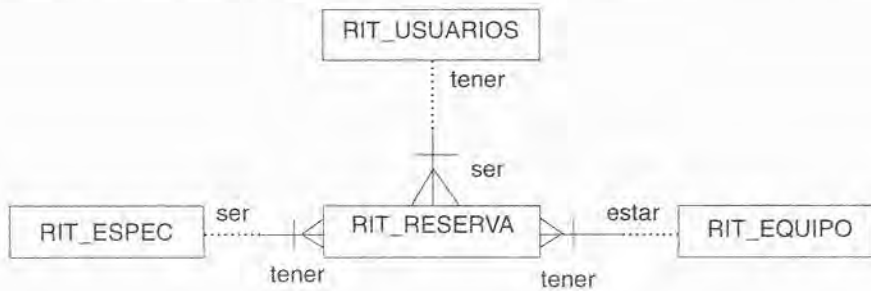
Para la entidad débil Rit_Reserva, se debe identificar la relación entre ésta y sus padres identificadores. Se debe realizar un seguimiento a la regla descrita en esa sección para poder deducir que hay una dependencia de inclusión entre Rit_Reserva con Rit_Espec, Rit_Equipos, y Rit_Usuario; de la siguiente manera:

W.X<<A.X, donde W es la entidad débil (Rit_Reserva), X es la llave primaria {Cod_Usuario, Sala}, y A, puede ser o bien sea Rit_Usuario, Rit_Espec ó Rit_Equipo. De acuerdo con la regla descrita en esta metodología, la cardinalidad para esta relación es 1:N y el modelo quedaría de la siguiente manera:



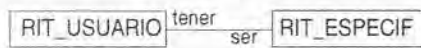
Como se puede observar, el modelo presenta relaciones muy rígidas (obligatorias) entre Rit_Reserva y Rit_Equipo, entre Rit_Reserva y Rit_Espec, y entre Rit_Reserva y Rit_Usuario. Por esta razón se debe hacer un análisis entre estas entidades, ya que no siempre un

usuario deberá tener una o más reservas, por esta razón se decidió no ser tan rígidos en nuestro caso práctico, colocando estas relaciones como opcionales (puede), quedando el modelo de la siguiente manera:



Después de realizado este paso, se entrará en la identificación de los tipos de relaciones. Para esto se debe realizar un seguimiento a la regla descrita en la sección 4.1, obteniendo una primera relación entre las entidades fuertes Rit_Espec y Rit_Usuarios, las cuales poseen la misma llave Cod_Usuario y, además existe una dependencia de inclusión entre Rit-Espec. {Cod_Usuario}<< Rit_Usuario.{Cod_Usuario} y Rit_Usuario.{Cod_Usuario}<< Rit_Espec.{Cod_Usuario}. Como se podrá observar, estas entidades cumplen con todos los requerimientos de esta regla, apareciendo así una relación **es-un** de cardinalidad mínima (1,1) entre las entidades anteriores, las cuales tienen el mismo grupo de instancias de datos para sus llaves primarias.

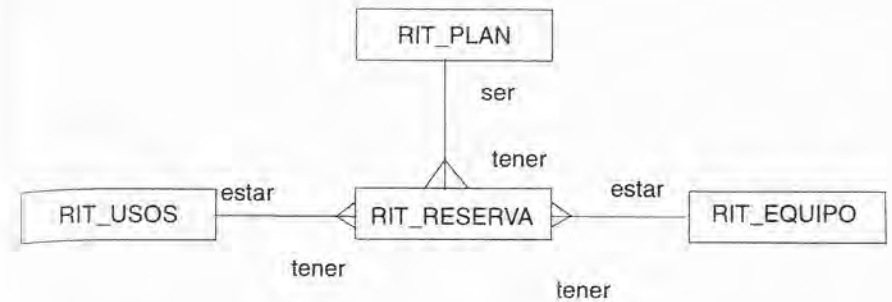
El modelo para esta relación sería:



Ahora se identificarán las relaciones binarias. Para esto, se debe hacer un seguimiento previo a la regla descrita en la sección 4.3; para poder realizar este paso se deben tener todas las llaves foráneas de una entidad y las dependencias de inclusión que se identificaron anteriormente. Las dependencias de inclusión que se identificaron para nuestro caso de estudio fueron las siguientes:

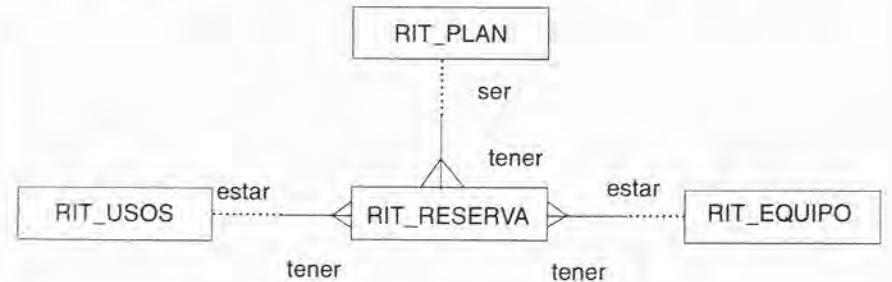
- Rit_Reserva.{Cod_uso}<< Rit_Usos. {Cod_Uso}
- Rit-Reserva.{Cod_Plan}<< Rit-Plan.{Cod_Plan}
- Rit_Reserva.{Cod_equipo}<< Rit_Equipos. {Cod-Equipo}

Donde la cardinalidad para este tipo de relación binaria identificada por una llave foránea es 1:N; quedando nuestro modelo de la siguiente manera:

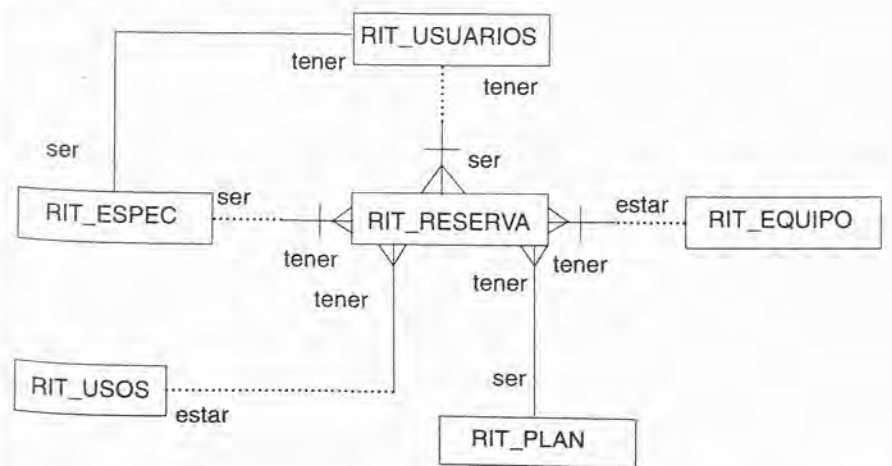


Al igual que en el modelo anterior, la regla infiere relaciones muy rígidas (obligatorias-debe), como la que se presenta entre Rit_Reserva y Rit_Equipo. Para este caso se haría la misma excepción

de no hacer nuestro sistema tan rígido, razón por la cual se decidió reemplazar esas relaciones obligatorias por opcionales (puede); quedando el modelo para este caso de la siguiente manera:



A continuación se ilustrará en conjunto el Modelo Entidad Relación para nuestro caso de estudio:



CONCLUSIONES

El presente trabajo nos ha permitido tener un conocimiento mucho más profundo sobre el manejo de las bases de datos relacionales, ya que al tratar la metodología de extracción, necesitamos contar con nuestro conocimiento y nuestro gran interés para aprender más sobre la forma correcta en que debe estar montada una base de datos para que permita el máximo rendimiento en la consecución de la información.

El reto fue muy grande porque entramos a tratar un tema en el que hay que tener no sólo el conocimiento adquirido en una materia sobre Bases de Datos o Ingeniería del Software, sino también una experiencia en el manejo o administración de una base de datos.

Nos queda la inquietud que en la etapa de análisis no se está teniendo en cuenta todo lo necesario para sacar un buen modelo de entidad-relación, ya que

la metodología de extracción nos ayuda a sacar elementos que de pronto nunca se han tratado en la materia de Ingeniería del Software, o si se han tratado, no se les ha dado la importancia que merecen para obtener en la etapa de diseño un buen modelo relacional.

BIBLIOGRAFIA

STOREY Veda C., i.e. *Reverse engineering or relational databases: Extraction of an EER model from a relational database*. Data & Knowledge Engineering 12 (1994). Págs. 107-142.

MARKOWITZ Victor M., i.e. *Identifying Extended Entity-Relationship Object Structures in Relational Schemas*. IEEE Transactions on Software Engineering. Vol. 6 No. 8. Agosto de 1990.

STOREY Veda C. *Relational database design based on the Entity-Relationship model*. Data & Knowledge Engineering 7 (1991). Págs. 47-83.

MODELO BASICO PARA EL MEJORAMIENTO DE LA CALIDAD DE VIDA DEL VIEJO RECLUSO, DURANTE SU PERMANENCIA EN LA CARCEL DEL DISTRITO DE CALI-VILLAHERMOSA

ESTHER JULIA ESPINOSA SALAZAR

Tecnóloga en Administración Hotelera y Comercio Exterior.

Tecnóloga en Gerontología, Universidad de San Buenaventura - Asistente de Relaciones Universitarias del ICESI.

JULIA MERCEDES LIBREROS GUTIERREZ

Gerontóloga, Universidad de San Buenaventura.

INTRODUCCION

El presente documento constituye un trabajo de investigación que nació de la necesidad de conocer la incidencia del sistema carcelario en el proceso de envejecimiento y la calidad de vida del individuo.

La investigación se llevó a cabo en la Cárcel del Distrito Judicial de Cali-Villahermosa, específicamente con la población vieja comprendida entre los 51 y 69 años de edad, la cual se encuentra agrupada en esta institución en un pasillo denominado de la Tercera Edad. Se pretende con esto encontrar nuevas alternativas de intervención gerontológica para este sector de la sociedad.

JUSTIFICACION

El incremento en el número de la población vieja es un fenómeno social y como tal debe estudiarse en todos los sectores de la sociedad, para conocer

las características particulares que inciden en este proceso.

El sistema carcelario es un subconjunto del gran sistema de leyes y normas sociales que rigen al individuo y que por lo tanto, se convierte en un factor importante de estudio en lo que atañe al envejecimiento de una nación.

Dado el carácter social de la Gerontología y teniendo en cuenta las recomendaciones que sobre cuestiones humanitarias hace la Asamblea Mundial de Viena sobre el envejecimiento, su numeral cuatro dice:

“Las cuestiones humanitarias se refieren más específicamente a la promoción del bienestar humano y la reforma social. Surgen de las necesidades y las condiciones humanas básicas de una sociedad y de la asignación de recursos financieros, humanos y naturales para erradicar aquellas condiciones que según el sistema de valores morales y