

**Unidad Didáctica**

# **BASE DE BATOS I**



# CONTENIDOS

|                                                                                                      |                                      |
|------------------------------------------------------------------------------------------------------|--------------------------------------|
| INTRODUCCIÓN.....                                                                                    | 6                                    |
| SOLUCIÓN TECNOLÓGICA PARA EL MANEJO DE INFORMACIÓN EN UN NEGOCIO.....                                | 7                                    |
| INTRODUCCIÓN A BASE DE DATOS.....                                                                    | 8                                    |
| CUESTIONARIO .....                                                                                   | <b>¡Error! Marcador no definido.</b> |
| ELABORACIÓN DEL PROYECTO DE BASE DE DATOS PARA MEJORAR EL SISTEMA DE INFORMACIÓN DE LA EMPRESA ..... | 14                                   |
| LA BASE DE DATOS CONCEPTOS GENERALES.....                                                            | 15                                   |
| 1. Conceptos Previos.....                                                                            | 15                                   |
| 2. Base de datos:.....                                                                               | 15                                   |
| 3. Tipos de base de datos.....                                                                       | 16                                   |
| 4. Características de una base de datos.....                                                         | 16                                   |
| 5. Modelo de datos.....                                                                              | 17                                   |
| 6. Usuarios y administradores de la base de datos.....                                               | 17                                   |
| PROCESO DE CREACIÓN DE UNA BASE DE DATOS.....                                                        | 19                                   |
| 1. Entradas y salidas del proceso de diseño de BD .....                                              | 20                                   |
| 2. Fases del diseño de BD.....                                                                       | 21                                   |
| 2.1. Análisis de requerimientos.....                                                                 | 21                                   |
| 2.1.1. Actividades de la determinación de Requerimientos.....                                        | 21                                   |
| 2.1.2. Métodos para la captura de datos.....                                                         | 22                                   |
| 2.1.2.1. Entrevistas: .....                                                                          | 22                                   |
| 2.1.2.2. Cuestionarios.....                                                                          | 24                                   |
| 2.1.2.3. Fuentes de información.....                                                                 | 25                                   |
| 2.1.2.4. Identificación de usuarios responsables:.....                                               | 26                                   |
| CUESTIONARIO .....                                                                                   | 28                                   |
| REPRESENTACION GRÁFICA DEL CONTENIDO E INTERACCIÓN DE LA INFORMACIÓN DE LA BASE DE DATOS.....        | 29                                   |
| DISEÑO CONCEPTUAL.....                                                                               | 30                                   |
| Modelo de datos Entidad- Relación .....                                                              | 30                                   |
| 1. Componentes del Modelo E-R.....                                                                   | 30                                   |
| 1.1. Conjunto de entidades.....                                                                      | 30                                   |
| 1.2. Atributos.....                                                                                  | 31                                   |
| 1.2.1. Dominio de un atributo.....                                                                   | 32                                   |
| 1.2.2. Tipos de atributos.....                                                                       | 32                                   |
| 1.3. Conjunto de relaciones.....                                                                     | 34                                   |
| 1.3.1. Tipos de relaciones.....                                                                      | 34                                   |

# CONTENIDOS

|                                                                                                                      |    |
|----------------------------------------------------------------------------------------------------------------------|----|
| 2. Restricciones.....                                                                                                | 34 |
| 3. Claves.....                                                                                                       | 38 |
| 3.1. Superclave.....                                                                                                 | 38 |
| 3.2. Clave primaria.....                                                                                             | 38 |
| 3.3. Clave candidata.....                                                                                            | 38 |
| 4. Diagrama Entidad – Relación.....                                                                                  | 39 |
| 4.1. Elementos gráficos:.....                                                                                        | 39 |
| 4.2. Complementación.....                                                                                            | 40 |
| CUESTIONARIO.....                                                                                                    | 45 |
| ESQUEMA RELACIONAL DE LA BASE DE DATOS DE LA EMPRESA.....                                                            | 46 |
| DISEÑO LÓGICO.....                                                                                                   | 47 |
| 1. Modelo Relacional.....                                                                                            | 47 |
| 1.1. Objetivos:.....                                                                                                 | 47 |
| 1.2. Estructura.....                                                                                                 | 47 |
| 1.2.1. Relaciones o relaciones.....                                                                                  | 47 |
| 1.2.2. Claves.....                                                                                                   | 49 |
| 1.3. Paso de entidad/relación de Chen al modelo relacional.....                                                      | 50 |
| 1.3.1. Transformación de las entidades fuertes:.....                                                                 | 50 |
| 1.3.2. Transformación de relaciones.....                                                                             | 50 |
| 1.4. Ejemplo empresa bancaria.....                                                                                   | 53 |
| CUESTIONARIO.....                                                                                                    | 55 |
| RESTRICCIONES EN EL DISEÑO DE LA BASE DE DATOS RELACIONAL DE LA<br>EMPRESA BASADOS EN LA NORMALIZACION DE DATOS..... | 56 |
| NORMALIZACIÓN.....                                                                                                   | 57 |
| 1. Problemas Del Esquema Relacional.....                                                                             | 57 |
| 2. Primera forma normal (1FN).....                                                                                   | 58 |
| 3. Dependencias funcionales.....                                                                                     | 58 |
| 3.1. Dependencia funcional.....                                                                                      | 59 |
| 3.2. Dependencia funcional completa.....                                                                             | 59 |
| 3.3. Dependencia funcional elemental.....                                                                            | 59 |
| 3.4. Dependencia funcional transitiva.....                                                                           | 59 |
| 4. Segunda forma normal (2FN).....                                                                                   | 59 |
| 5. Tercera forma normal (3FN).....                                                                                   | 60 |
| 6. Forma normal de Boyce-Codd (FNBC o BCFN).....                                                                     | 61 |
| 7. Cuarta forma normal (4FN). Dependencias Multivaluadas.....                                                        | 62 |
| 8. Quinta forma normal (5FN).....                                                                                    | 64 |
| 9. Forma normal de dominio clave (FNDC).....                                                                         | 65 |
| Cuestionario.....                                                                                                    | 67 |

# CONTENIDOS

|                                                                                                                         |     |
|-------------------------------------------------------------------------------------------------------------------------|-----|
| REPORTES DE INFORMACIÓN DE UN NEGOCIO .....                                                                             | 68  |
| 1. Operaciones fundamentales .....                                                                                      | 69  |
| 1.1. La operación selección.....                                                                                        | 69  |
| 1.2. La operación proyección .....                                                                                      | 72  |
| 1.3. La operación unión.....                                                                                            | 74  |
| 1.4. La operación diferencia de conjuntos .....                                                                         | 75  |
| 1.5. La operación producto cartesiano .....                                                                             | 76  |
| 1.6. Operación renombramiento.....                                                                                      | 78  |
| CUESTIONARIO .....                                                                                                      | 81  |
| DIAGRAMACIÓN DEL MODELO DE BASE DE DATOS DEL NEGOCIO USANDO<br>COMO HERRAMIENTA TECNOLÓGICA ERWIN .....                 | 82  |
| INTRODUCCIÓN A Erwin 7.1 .....                                                                                          | 83  |
| 1. Conceptos básicos. ....                                                                                              | 83  |
| 2. Descripción del entorno .....                                                                                        | 84  |
| 3. Creando Entidades y Atributos .....                                                                                  | 89  |
| 3.1. Construyendo Entidades .....                                                                                       | 89  |
| 3.2. Editar entidades: .....                                                                                            | 90  |
| 4. Elaboramos: Modelo Cine.....                                                                                         | 91  |
| 4.1. Entidades y atributos.....                                                                                         | 91  |
| 4.2. Creación de llaves.....                                                                                            | 92  |
| 4.3. Relaciones .....                                                                                                   | 95  |
| 4.4. Relaciones de Muchos a Muchos.....                                                                                 | 98  |
| 5. Creando Áreas de Negocio y Stored Display .....                                                                      | 101 |
| 6. Optimizando el Modelo de datos.....                                                                                  | 105 |
| 6.1. Generalización Jerárquica.....                                                                                     | 105 |
| 6.2. Integridad Referencial (RI) .....                                                                                  | 106 |
| 7. Modelamiento Físico.....                                                                                             | 107 |
| 7.1. Seleccionar el servidor de la base de datos. ....                                                                  | 108 |
| 7.2. Propiedades de columna.....                                                                                        | 109 |
| 7.3. Asignación de tipos de datos.....                                                                                  | 109 |
| CUESTIONARIO .....                                                                                                      | 110 |
| DISEÑO DE LA BASE DE DATOS RELACIONAL DE LA EMPRESA UTILIZANDO EL<br>GESTOR DE BASE DE DATOS MICROSOFT ACCES 2013 ..... | 111 |
| INTRODUCCION AL DISEÑO FISICO DE LA BASE DE DATOS.....                                                                  | 112 |
| 1. Sistema Gestor De Base De Datos.....                                                                                 | 112 |
| 2. Las funciones de un Sistema Gestor de Bases de Datos (SGBD) .....                                                    | 112 |
| 3. Microsoft Access.....                                                                                                | 112 |
| 3.1. Crear una base de datos de escritorio.....                                                                         | 113 |

# CONTENIDOS

|          |                                              |     |
|----------|----------------------------------------------|-----|
| 3.1.1.   | Tablas.....                                  | 115 |
| 3.1.2.   | Panel de navegación .....                    | 117 |
| 3.1.3.   | Relaciones entre tablas. ....                | 120 |
| 3.1.3.1. | Clave principal:.....                        | 120 |
| 3.1.3.2. | Tipos de relaciones .....                    | 122 |
| 3.1.3.3. | Proceso de relación .....                    | 122 |
| 3.1.3.4. | Opciones para la relación entre tablas ..... | 124 |
| 3.1.3.5. | Trabajar con relaciones.....                 | 125 |
| 3.1.3.6. | Eliminar relaciones.....                     | 126 |
| 3.1.3.7. | Dependencias del objeto.....                 | 127 |
| 3.1.4.   | Insertar datos en la tabla. ....             | 128 |
| 3.1.5.   | Consultas .....                              | 129 |
| 3.1.6.   | Las consultas de selección .....             | 129 |
| 3.1.7.   | Diseño de consultas. ....                    | 129 |
| 3.1.8.   | Guardar consultas .....                      | 134 |
| 3.1.9.   | Ejecución y resultados.....                  | 134 |
|          | CUESTIONARIO .....                           | 136 |
|          | BIBLIOGRAFIA.....                            | 137 |

# INTRODUCCIÓN

Las organizaciones o empresas, sin distinguirse en tamaño, necesitan la información, para mejorar sus procesos de negocio es por ello que no se almacena volúmenes exagerados de esta sino que se hace un filtrado; que dándose con la que pueda servir como aliada estratégica; creando así una base de datos que con participación de tecnología y personal capacitado le permita consolidarse en el mercado según la visión establecida.

Este manual tiene por objetivo incursionar a los estudiantes en este mundo de la información brindándole los conocimientos y ejemplos prácticos que le permitan establecer un correcto análisis de la información de los procesos de negocio y requerimientos que puedan tener las empresas u organizaciones, y obtener un diseño óptimo que albergue la solución de los requerimientos; estos conceptos incluyen aspectos de análisis y diseño de bases de datos por ser el primer curso de base de datos para nivel técnico; así mismo mostraremos ejemplos que se reflejen al trabajo actual y real de las empresas.

Este manual está estructurado en las siguientes actividades:

**Actividad 1:** Mostraremos una visión amplia de la importancia de la información y por ende de la base de datos como estrategia en una empresa; así mismo de lo que necesitamos para su creación y mantenimiento.

**Actividad 2:** Conoceremos conceptos generales de base de datos y, aprenderemos a realizar la captura de datos que nos servirán en la primera fase de la creación de una base de datos; determinaremos los requerimientos de la empresa.

**Actividad 3:** Se detallara la información correspondiente al diseño conceptual e interacción de la información de una empresa con ayuda del **Modelo Entidad- Relación** y **El Diagrama Entidad-Relación**.

**Actividad 4:** Desarrollaremos el diseño lógico de la base de datos relacional para ello usaremos en **Modelo Relacional y sus principales características**.

**Actividad 5:** Se detallaran los conceptos de **Normalización** de datos que serán aplicados a nuestro esquema relacional; para establecer un buen diseño.

**Actividad 6:** Detallaremos las operaciones básicas del **Álgebra Relacional**.

**Actividad 7:** Utilizaremos Erwin para la diagramación del modelo de datos lógico y físico de la base de datos de la empresa en nivel básico.

**Actividad 8:** Daremos paso a la introducción del diseño físico de la base de datos para lo cual utilizaremos como sistema gestor de base de datos **Microsoft Access 2013**;



# ACTIVIDAD I

## **SOLUCIÓN TECNOLÓGICA PARA EL MANEJO DE INFORMACIÓN EN UN NEGOCIO**

**OBJETIVO:** Se pretende que el estudiante conozca una visión general de la importancia de los sistemas de información para las empresas, y las ventajas que ofrecería una base de datos para esta.

## INTRODUCCIÓN A BASE DE DATOS



### Sistema de información (SI)

"Es un conjunto de componentes que interaccionan entre sí para alcanzar un fin determinado, el cual es satisfacer las necesidades de información de dicha organización. Estos componentes pueden ser **personas, datos, actividades o recursos materiales en general**, los cuales procesan la **información** y la distribuyen de manera adecuada, buscando satisfacer las necesidades de la organización".



A cartoon illustration of a man with brown hair, wearing a white shirt and a brown tie. He has a speech bubble above his head. The speech bubble is blue with a white border and contains the text: "Toda la información es importante para la empresa." The man has a slightly confused or questioning expression on his face.

Toda la información es importante para la empresa.

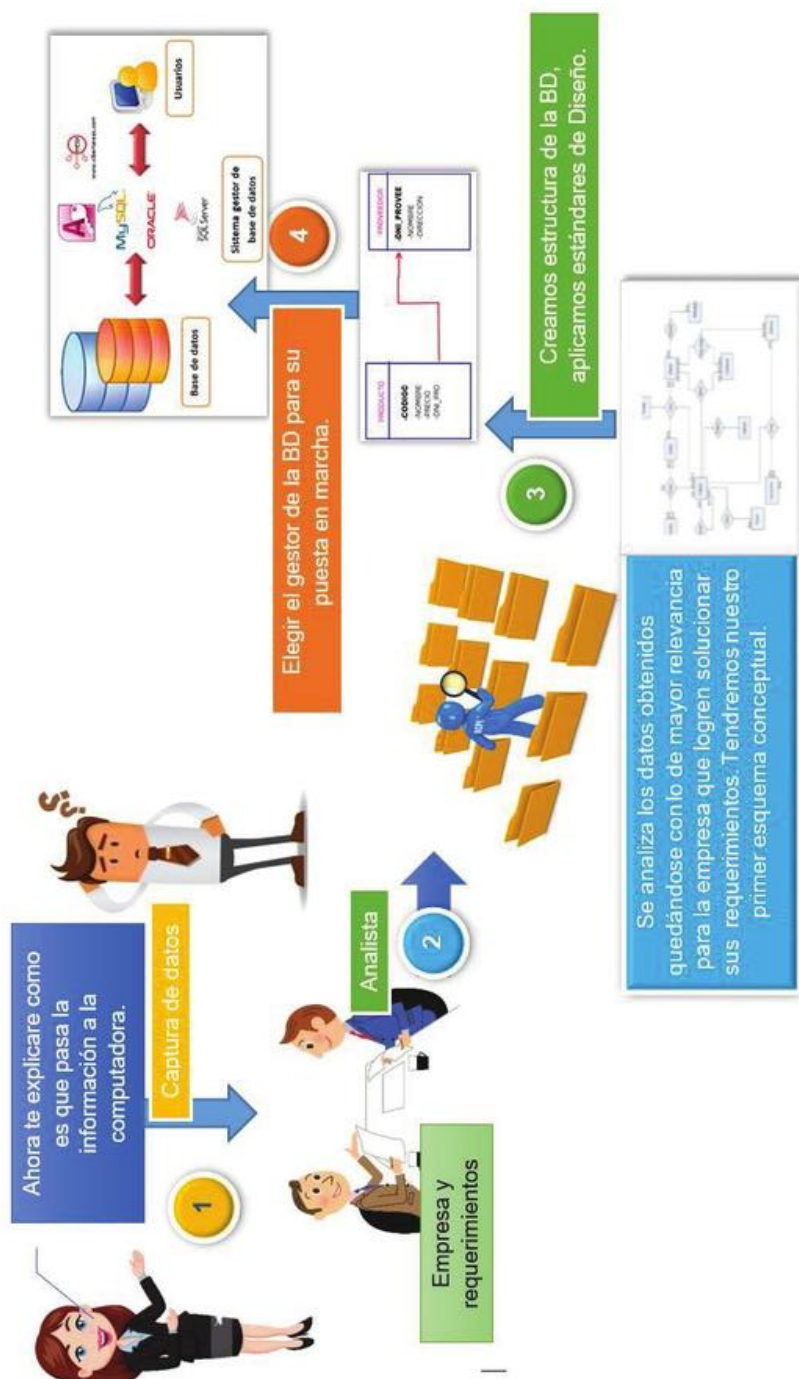
A cartoon illustration of a woman with long brown hair, wearing a black business suit. She has a speech bubble above her head. The speech bubble is yellow with a white border and contains the text: "No, solo aquella que le permite agilizar sus procesos, tener un alto nivel de competitividad y posibilidades de desarrollo." The woman has a confident and explanatory expression on her face.

No, solo aquella que le permite agilizar sus procesos, tener un alto nivel de competitividad y posibilidades de desarrollo.

A cartoon illustration of a man with brown hair, wearing a white shirt and a brown tie. He has a speech bubble above his head. The speech bubble is yellow with a white border and contains the text: "Ese conjunto de información debe ser organizada, relacionada y almacenada, para encontrarla y utilizarla cuando la empresa crea conveniente; eso es una BASE DE DATOS." The man has a slightly confused or questioning expression on his face.

Ese conjunto de información debe ser organizada, relacionada y almacenada, para encontrarla y utilizarla cuando la empresa crea conveniente; eso es una **BASE DE DATOS.**







CUESTIONARIO

Investigación:

1. En qué tipo de empresas podemos tener una base de datos.
2. Menciona las ventajas de tener base de datos en las empresas.
3. ¿Qué recomendarías aquellas empresas que aún no optaron por una BD automatizada?

---

---

---



## ACTIVIDAD II

### **ELABORACIÓN DEL PROYECTO DE BASE DE DATOS PARA MEJORAR EL SISTEMA DE INFORMACIÓN DE LA EMPRESA**

**OBJETIVO:** Dar a conocer los conceptos preliminares de base de datos, y métodos adecuados para realizar una correcta toma de información de un negocio real que le permita analizar los requerimientos y conocer los procedimientos del negocio.

## LA BASE DE DATOS CONCEPTOS GENERALES

### 1. Conceptos Previos.

#### Datos

Representación simbólica (numérica, alfabética, etc.) de un atributo de una entidad. Un dato no tiene valor semántico (sentido) en sí mismo, pero al ser procesado puede servir para realizar cálculos o tomar decisiones.

Los datos son susceptibles de ser comprimidos, encriptados, transmitidos y almacenados.

#### Información

La información no es un dato conjunto cualquiera. Es una colección de hechos significativos y pertinentes, para el organismo u organización que los percibe. La definición de información es la siguiente: Información es un conjunto organizado de datos significativos y pertinentes que describan sucesos o entidades, haciendo uso racional es la base del Conocimiento.

### 2. Base de datos:

Es un contenedor que permite almacenar la información de forma ordenada con diferentes propósitos y usos. Por ejemplo, en una base de datos se puede almacenar información de diferentes departamentos (Ventas, Recursos Humanos, Inventarios, entre otros). El almacenamiento de la información por sí sola no tiene un valor, pero si combinamos o relacionamos la información con diferentes departamentos nos puede dar valor.

#### **Aplicaciones de los sistemas de bases de datos:**

En la actualidad las bases de datos son ampliamente usadas en entornos comerciales. Las siguientes son algunas de sus aplicaciones más representativas:

- **Banca.** Para información de los clientes, cuentas y préstamos, y transacciones bancarias.
- **Líneas aéreas.** Para reservas e información de planificación. Las líneas aéreas fueron de los primeros en usar las bases de datos de forma distribuida geográficamente (los terminales situados en todo el mundo accedían al sistema de bases de datos centralizado a través de las líneas telefónicas y otras redes de datos).
- **Universidades.** Para información de los estudiantes, matrículas de las asignaturas y cursos.
- **Transacciones de tarjetas de crédito.** Para compras con tarjeta de crédito y generación mensual de extractos.
- **Telecomunicaciones.** Para guardar un registro de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.
- **Finanzas.** Para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.
- **Ventas.** Para información de clientes, productos y compras.
- **Producción.** Para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos en almacenes y pedidos de elementos.
- **Recursos humanos.** Para información sobre los empleados, salarios, impuestos y beneficios, y para la generación de las nóminas.

Como nos daremos cuenta utilizamos base de datos en todas las empresas ya que representan una ayuda sustancial a la mejora del negocio.

### 3. Tipos de base de datos.

Existen muchas empresas con diferentes servicios y dependiendo de estos será el tipo de procesamiento que se le dará a la información, esto determinará el tipo de base de datos a utilizar.

Existen diferentes tipos de bases de datos pero las más comunes son las OLTP y OLAP.

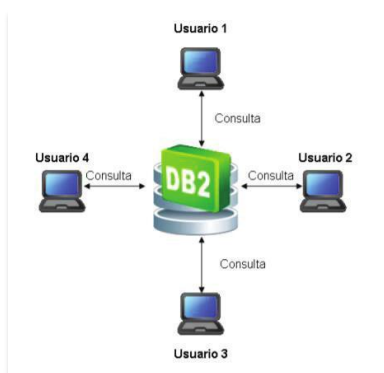
#### • Las bases de datos de tipo OLTP (On Line Transaction Processing)

también son llamadas bases de datos dinámicas lo que significa que la información se modifica en tiempo real, es decir, se insertan, se eliminan, se modifican y se consultan datos en línea durante la operación del sistema. Un ejemplo es el sistema de un ventas donde se van registrando cada uno de los artículos que el cliente está comprando y a su vez el sistema va actualizando el inventario.



#### • Las bases de datos de tipo OLAP (On Line Analytical Processing)

también son llamadas bases de datos estáticas lo que significa que la información en tiempo real no es afectada; solo se realizan consultas sobre los datos ya existentes para el análisis y toma de decisiones. Este tipo de bases de datos son implementadas en Business Intelligence para mejorar el desempeño de las consultas con grandes volúmenes de información.



### 4. Características de una base de datos.

- Independencia lógica y física.
- Redundancia mínima.
- Accesos concurrentes por parte de múltiples usuarios.
- Integridad de los datos.
- Seguridad de accesos auditoria.
- Respaldo y recuperación.

## 5. Modelo de datos.

Es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. Para ilustrar el concepto de un modelo de datos, describimos dos modelos de datos en este apartado: el modelo entidad relación y el modelo relacional.

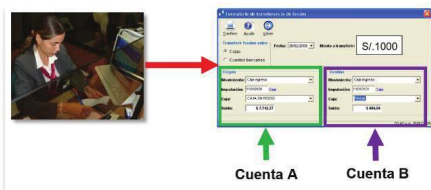
- **Modelo entidad-relación:** El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos.
- **Modelo relacional:** En el modelo relacional se utiliza un grupo de relaciones para representar los datos y las relaciones entre ellos. Cada relación está compuesta por varias columnas, y cada columna tiene un nombre único.
- **Otros modelos de datos:**
  - El **modelo orientado a objetos** se puede observar como una extensión del modelo E-R con las nociones de encapsulación, métodos (funciones) e identidad de objeto.
  - El **modelo de datos relacional orientado a objetos** combina las características del modelo de datos orientado a objetos y el modelo de datos relacional.
  - Los **modelos de datos semiestructurados** permiten la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos. Esto es diferente de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos. El lenguaje de marcas extensible (XML, eXtensible Markup Language) se usa ampliamente para representar datos semiestructurados.
  - Históricamente, otros dos modelos de datos, **el modelo de datos de red y el modelo de datos jerárquico, precedieron al modelo de datos relacional**. Estos modelos estuvieron ligados fuertemente a la implementación subyacente y complicaban la tarea del modelado de datos. Como resultado se usan muy poco actualmente, excepto en el código de bases de datos antiguo que aún está en servicio en algunos lugares.

## 6. Usuarios y administradores de la base de datos.

Siempre debemos tener en claro que lo primordial en una base de datos es que nos permita recuperar y almacenar información; por ello que contamos con personas que trabajan con una base de datos a las que podemos denominar usuarios de bases de datos o administradores de bases de datos.

- **Usuarios normales.** Son usuarios no sofisticados que interactúan con el sistema mediante la invocación de alguno de los programas de aplicación permanentes que se ha escrito previamente.

Por ejemplo, un cajero bancario que necesita transferir S/.1000.00 de la cuenta “A” a la cuenta “B” invoca un programa llamado transferir.



Este programa pide al cajero el importe de dinero a transferir, la cuenta de la que el dinero va a ser transferido y la cuenta a la que el dinero va a ser transferido. La **interfaz de usuario normal** para los usuarios normales es una interfaz de formularios, donde el usuario puede rellenar los campos apropiados del formulario. Los usuarios normales pueden también simplemente leer informes generados de la base de datos.

- **Programadores de aplicaciones.** Son profesionales informáticos que escriben programas de aplicación. Los programadores de aplicaciones pueden elegir entre muchas herramientas para desarrollar interfaces de usuario.
- **Los usuarios sofisticados:** Interactúan con el sistema sin programas escritos. En su lugar, ellos forman sus consultas en un lenguaje de consulta de bases de datos. Cada una de estas consultas se envía al procesador de consultas, cuya función es transformar instrucciones LMD (Lenguaje de manipulación de datos) a instrucciones que el gestor de almacenamiento entienda. Los analistas que envían las consultas para explorar los datos en la base de datos entran en esta categoría.

Por ejemplo, un analista puede ver las ventas totales por región (por ejemplo, norte, sur, este y oeste), o por producto, o por una combinación de la región y del producto (es decir, las ventas totales de cada producto en cada región). una herramienta muy utilizada es OLAP (procesamiento analítico en línea).

- **Usuarios especializados.** Son usuarios sofisticados que escriben aplicaciones de bases de datos especializadas que no son adecuadas en el marco de procesamiento de datos tradicional. Entre estas aplicaciones están los sistemas de diseño asistido por computador, sistemas de bases de conocimientos y sistemas expertos, sistemas que almacenan los datos con tipos de datos complejos (por ejemplo, datos gráficos y datos de audio) y sistemas de modelado del entorno.
- **Administrador de la base de datos:** Una de las principales razones de usar SGBDs (sistema de gestión de base de datos) es tener un control centralizado tanto de los datos como de los programas que acceden a esos datos. La persona que tiene este control central sobre el sistema se llama administrador de la base de datos (ABD).

Las funciones del ABD incluyen las siguientes:

- **Definición del esquema.** El ABD crea el esquema original de la base de datos escribiendo un conjunto de instrucciones de definición de datos en el LDD (lenguaje de definición de datos).
- **Definición de la estructura y del método de acceso.**
- **Modificación del esquema y de la organización física.** Los ABD realizan cambios en el esquema y en la organización física para reflejar las necesidades cambiantes de la organización, o para alterar la organización física para mejorar el rendimiento.
- **Concesión de autorización** para el acceso a los datos. La concesión de diferentes tipos de autorización permite al administrador de la base de datos determinar a qué partes de la base de datos puede acceder cada usuario. La información de autorización se mantiene en una estructura del sistema especial que el sistema de base de datos consulta cuando se intenta el acceso a los datos en el sistema.

- **Mantenimiento rutinario.** Algunos ejemplos de actividades rutinarias de mantenimiento del administrador de la base de datos son:
  - **Copia de seguridad periódica de la base de datos**, bien sobre cinta o sobre servidores remotos, para prevenir la pérdida de datos en caso de desastres como inundaciones.
  - **Asegurarse de que haya suficiente** espacio libre en disco para las operaciones normales y aumentar el espacio en disco según sea necesario.
  - **Supervisión de los trabajos que se ejecuten en la base de datos** y asegurarse de que el rendimiento no se degrada por tareas muy costosas iniciadas por algunos usuarios.

### PROCESO DE CREACIÓN DE UNA BASE DE DATOS.

Luego de ver algunos conceptos importantes y necesarios para la explicación de esta actividad, nos concentraremos en el proceso de diseño de una base de datos y sus respectivas faces.



## 1. Entradas y salidas del proceso de diseño de BD



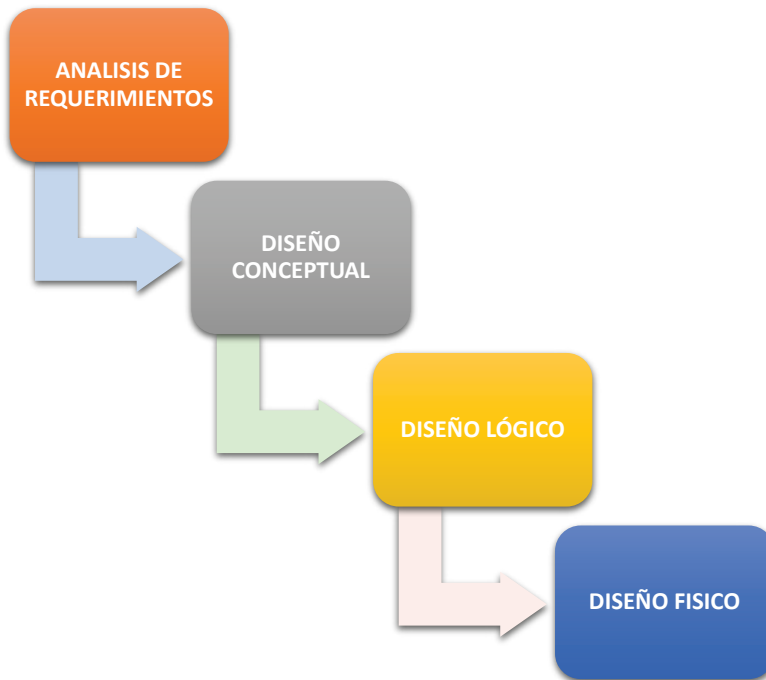
### a) Entradas.

- **Requisitos de información y objetivos.** Entrevistas con los usuarios. Análisis de los documentos a generar. Objetivos de la organización.
- **Requisitos de procesos.** Características que deben cumplir las aplicaciones, por ej. Tiempo de respuesta.
- **Especificaciones del SGBD. Modelo de datos soportado.** Características de rendimiento, seguridad, lenguajes...
- **Configuración del equipo físico y del SO.** Influirán en el diseño físico y ajuste de la BD.

### b) Salidas.

- **Estructura lógica de datos.** Esquema conceptual. Esquema lógico en el modelo soportado por el SGBD. Vistas de usuario.
- **Estructura de almacenamiento.** Esquema interno con especificaciones físicas como particiones, definiciones de espacio, índices, cluster..
- **Normativa de explotación.** Confidencialidad, seguridad e integridad para la BD.
- **Especificaciones para los programas.** Características que no pueden ser recogidas en el esquema.

## 2. Fases del diseño de BD



### 2.1. Análisis de requerimientos.

Esta etapa tiene por finalidad en conocer el mundo real que vamos analizar, en términos comunes diremos que es momento de identificar y describir los procesos y usuarios que interviene en el negocio de la empresa; y los requerimientos que esta desea cumplir que lo veremos reflejados en el producto final.

Entonces el primer paso del **analista es comprender la situación actual**.

#### 2.1.1. Actividades de la determinación de Requerimientos

| Anticipación de Requerimientos                                                                                                                                                     | Investigación de Requerimientos                                                                                                                 | Especificación de Requerimientos                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Prever las características de sistema con base a la experiencia previa. Esto puede llevar al analista a investigar áreas y aspectos que de otra forma no serían tomados en cuenta. | Estudio y documentación del sistema actual utilizando para ello técnicas para hallar hechos, análisis de flujo de datos y análisis de decisión. | Análisis de los datos que describen el sistema para determinar qué tan bueno es su rendimiento, qué requerimientos deben satisfacer y las estrategias para alcanzarlos. |
| La experiencia permite anticipar requerimientos para el nuevo sistema.                                                                                                             | Es la actividad más importante                                                                                                                  | Implica un riesgo ya que dependiendo de la calidad de requerimientos afectará al nuevo sistema.                                                                         |

**Los analistas estructuran su investigación en base a 4 preguntas:**

1. ¿Cuál es el proceso básico de la empresa?
2. ¿Qué datos utiliza o produce este proceso?
3. ¿Qué frecuencia y volumen del proceso existe?

4. ¿Qué controles utiliza para su realización?

**Estudiamos por separado cada una de estas cuatro preguntas:**

**¿Cuál es el proceso básico de la empresa?**

Las siguientes preguntas se utilizan para adquirir la comprensión necesaria:

- ¿Cuál es la finalidad de esta actividad en la empresa?
- ¿Qué pasos se siguen para llevarla a cabo?
- ¿Dónde se realizan estos pasos?
- ¿Quiénes los realizan?
- ¿Cuánto tiempo tardan en efectuarlos?
- ¿Con cuanta frecuencia lo hacen?
- ¿Quiénes emplean la información resultante?

**¿Qué datos utiliza o produce este proceso?**

Este paso consiste en detectar qué datos se utilizan para llevar a cabo cada actividad.

**¿Qué frecuencia y volumen de proceso existe?**

Los analistas deben investigar con cuanta frecuencia se repite una actividad. Esto cambia mucho dependiendo de la actividad ya que por ejemplo el pago de la nómina se repite mensualmente o semanalmente pero el pago de impuestos es anualmente.

La manera más fácil de obtener esta información es identificar el objetivo de la actividad, es decir, cuál es la causa de la actividad.

El volumen de los procesos puede aumentar el tiempo de realización de las actividades, es decir la cantidad total de pasos que puede constar una actividad puede generar problemas aun ocurriendo con poca frecuencia.

**¿Qué controles utiliza para su realización?**

La falta o debilidad de los controles es un descubrimiento importante en cualquier investigación del sistema.

El analista debe examinar los métodos de control preguntando:

- ¿Quién se encarga de comparar lo realizado con los estándares?
- ¿Cómo se detectan los errores?
- ¿Cómo se corrigen los errores?

**2.1.2. Métodos para la captura de datos**

Te estarás preguntando como vamos a recolectar toda esta información; bien para ello haremos usos de algunos métodos para la captura de datos:

**2.1.2.1. Entrevistas:**

Este es el método más corriente para recoger información del usuario. Es un proceso continuo utilizado por el analista para

construir gradualmente un modelo del sistema y para obtener conocimientos sobre los problemas del sistema. Hay dos factores clave en la realización de entrevistas. El primero es elegir a la persona a entrevistar. Es importante, así que el analista debe asegurarse de que se considera a todas las personas clave dentro del entorno del estudio. El siguiente factor es encontrar el camino correcto para dirigir una entrevista individual. Aquí se deben considerar las relaciones humanas. El entrevistador debe establecer algunos contactos con el entrevistado a fin de asegurar la cooperación necesaria para obtener todos los factores relevantes.

Con los usuarios que participan en los procesos de empresa organización.

#### a) El plan de entrevista:

El plan de la entrevista específica:

- El usuario a entrevistar
- La secuencia en la que será entrevistado
- El plan de entrevista para cada usuario

Un plan de entrevista se basa generalmente en la estructura organizativa actual, que está formada por unidades organizativas.

El analista utiliza los términos del proyecto de referencia para seleccionar las áreas organizativas que caen dentro de los límites de estudio del sistema y que se verán afectadas probablemente por cualquier nuevo sistema.

Normalmente, es deseable comenzar las entrevistas por el nivel más alto de la organización para obtener el soporte y la cooperación de la dirección antes de comenzar a examinar actividades organizativas particulares o sugerir nuevas soluciones.

Así, pues, la preparación de las entrevistas es esencial. Se debe tener una idea de lo que se pretende conseguir con la entrevista y formular preguntas directas para obtener esa información.

No se debe esperar obtener toda la información necesaria de un usuario en el curso de una entrevista. Normalmente, hay una serie de dos, tres o más entrevistas con un usuario. Esta primera entrevista va seguida de un número de entrevistas recogidas de factores para enterarse de todo lo importante que conoce el usuario. A continuación se pueden hacer una o más entrevistas para verificar esos factores y cualquier otro modelo desarrollado por el analista, y para conseguir información adicional que complete el estudio del analista.

## b) Contenido de la entrevista

- **Las entrevistas** deben efectuarse de forma organizada y amistosa. El entrevistador siempre será cortés y respetará la oposición y necesidades del usuario. Es importante no imponer soluciones a los usuarios, sino el papel de asesor. La jerga informática no se debe utilizar para impresionar al usuario, los entrevistadores deben explicar las limitaciones del ordenador en términos cotidianos y describir al usuario cómo le puede ayudar en su trabajo.
- **Los entrevistadores** deberán asegurarse de obtener toda la información necesaria de la entrevista. Para eso ayuda poner en conocimiento del usuario la información que se pretende obtener con la entrevista. El entrevistador buscará entonces esa información gradualmente y formular preguntas precisas y concretas.
- **Una entrevista normal** procedería según los siguientes pasos. **Primero**, se establecen los procedimientos de entrevistas. Esto incluye la duración y lo que se piensa obtener, así como conseguir permiso para tomar apuntes y notas durante la entrevista. Es una buena idea hacer esto, pues el entrevistado conoce de antemano lo que se espera de la entrevista y se sentirá más predispuesto a colaborar.
- **Una vez acabados los preliminares**, puede comenzar el cuerpo de la entrevista. Normalmente, es una buena idea empezar confirmando la información obtenida en entrevistas anteriores o en alguna investigación. Esto sirve para situar al entrevistado y ayuda a encontrar errores en los datos. Una vez que se está conforme, se sigue con más detalle cualquier punto relevante.
- **La entrevista se terminará** resumiendo lo obtenido de ella y confirmándolo. Finalmente, es buena idea convenir la fecha para la siguiente entrevista, si se considera necesaria.

## 2.1.2.2. Cuestionarios

El cuestionario contiene todas las preguntas que el usuario debe responder para proporcionar la información que busca el analista. El cuestionario se envía al usuario y el analista analiza las respuestas.

La experiencia sugiere que estos cuestionarios no son normalmente buenos sustitutos de las entrevistas. Por lo general, las preguntas como 'describa todos sus trabajos' o '¿cuáles son los componentes principales del sistema?' no son efectivas.

Estas preguntas normalmente no se responderán completamente y en general expresarán sucesos recientes. Así, la respuesta a '**describa sus trabajos**' incluirá generalmente las cosas que el usuario ha hecho el último día.

Para establecer las actividades normalmente es necesario que durante un periodo largo de tiempo se sucedan una serie de

preguntas, y esto es mejor hacerlo mediante una entrevista, en vez de con un cuestionario enorme.

**Los cuestionarios**, sin embargo, se utilizan cuando se busca la misma información en usuarios distintos. Es el caso de información de naturaleza cuantitativa. Un cuestionario con esta pregunta es fácil enviarlo a todos los vendedores de la organización.

Los cuestionarios se utilizan también como complemento de otras técnicas.

Se usan para recoger datos numéricos u obtener opiniones relativamente simples de un número de personas, pero no son efectivos para búsquedas detalladas ni para identificar problemas o soluciones del sistema. Las entrevistas son las más provechosas para este propósito.

### 2.1.2.3. Fuentes de información.

Cada fuente da información distinta y requiere un método de búsqueda diferente para conseguirla. Algunas de las fuentes más comunes y métodos de búsqueda apropiados para ellas se exponen seguidamente.

- **Usuarios del sistema:**  
Los usuarios del sistema son normalmente la primera fuente de información a investigar por el analista. De los usuarios es posible extraer información de las actividades del sistema existente y determinar los objetivos del usuario y sus necesidades. Aquí, los métodos usuales son las entrevistas y, a veces, cuestionarios.
- **Formularios y documentos**  
Los formularios y documentos son fuentes de información utilizadas para los diagramas de flujos de datos y transacciones. El método de búsqueda comienza con la obtención por parte del analista de una lista de tales documentos de usuario del sistema, para, a través de ellos, encontrar los datos elementales y sus estructuras de datos. En ese punto, es corriente un control de duplicidad de datos y de consistencia de nombres para asegurar que el mismo dato no aparece con dos nombres distintos.
- **Programas**  
Los programas se pueden utilizar para determinar los detalles de las estructuras de datos o de los procesos. Los métodos de búsqueda son generalmente laboriosos y suponen la lectura del programa o su documentación y, a veces, su ejecución con datos de muestra para ver lo que hace o si algo está en desacuerdo con el interfaz actual del usuario.
- **Manual de procedimiento**  
Los manuales de procedimiento especifican lo que hace el personal de una organización. Los puede utilizar el analista

para determinar el detalle las actividades del usuario. Esta información adquiere relevancia en el diseño detallado. El método sigue necesitando el examen detallado de las técnicas de la metodología del diseño aplicable.

- **Informes**

Esta fuente indica las clases necesarias para el usuario. Se pueden utilizar como base para las entrevistas con el usuario y así determinar cualquier nuevo requerimiento de salida que pueda tener. Es muy improbable que una de estas fuentes pueda, por sí sola, suministrar toda la información que necesita un analista. Si se parte de un sistema existente, con toda certeza se podrá obtener información sobre la organización de la mayoría de las fuentes, si no de todas. El procedimiento de búsqueda determina cómo llevar a cabo una búsqueda revisando esas fuentes.

#### **2.1.2.4. Identificación de usuarios responsables:**

- ♦ **Alta dirección.**

- Objetivos y metas corporativas.
- Visión de funciones importantes
- Evolución futura.
- Establecer prioridades.

- ♦ **Mandos intermedios.**

- Objetivos detallados.
- Identificar usuarios intermedios

- ♦ **Usuarios operacionales.**

- Requerimientos detallados.
- Procedimientos.
- Informes.
- Formularios.

Ejemplo: La empresa Computer G&G desea estructurar su información, el equipo de análisis ha preparado un conjunto de preguntas para entrevistar al dueño del negocio.

#### **Análisis de requerimientos de una base de datos; entrevista.**

¿Cuál es su nombre y cargo que ocupa?

Rpta: Willy Fernandez, dueño de la empresa.

¿Cuál es el rubro que tiene su empresa?

Rpta: Venta de computadoras y accesorios.

¿Qué tipos de productos vende?

Rpta: Computadoras, laptop, Tablet, grabadoras ópticas, teclado, mouse, etc.

¿A qué público va dirigido su negocio?

Rpta: Público en general

¿Qué problemas ha tenido en su negocio con respecto a la información?

Rpta: El no saber:

- Cuántos productos vendimos y nuestras ganancias del día.
- Cuántos productos tenemos en existencia.

¿Por qué quiere resolver estos problemas?

Rpta: Quiero tener mejor organización con las entradas y salidas de productos, que los vendedores sepan en el instante con cuantos productos contamos y puedan brindar un servicio más eficiente a nuestros clientes lo cual significaría mejores ingresos y el reconocimiento de nuestra empresa.

¿Tiene conocimientos en computación?

Rpta: Conocimiento básico.

¿En qué ambiente se utilizara el sistema?

Rpta: La tienda tiene una habitación disponible es pequeña pero esta alejada del personal y público.

¿Qué datos de los productos son importantes?

Rpta: El tipo de producto, para qué sirve el producto, si está sujeto a garantía, en que día y cantidad ingreso a la tienda, la marca, su proveedor.

¿Qué datos requieres de los proveedores?

Rpta: El nombre del representante, su dirección, teléfono, datos con lo que los pueda localizar sin contra tiempos.

¿Con respecto a los clientes que le gustaría saber de ellos?

Rpta: Me gustaría tener una lista de clientes con los datos que me permitan contactar con ellos para enviar ofertas de productos o avisarle sobre el estado del producto cuando lo dejan en taller de soporte técnico.

Observaciones:

- Es un negocio pequeño.
- Tiene dos años de apertura.
- No cuenta con libro de entrada y salida de datos.
- Tiene posibilidades de adquirir computadora, impresora y lector óptico.



*Que una buena sesión informativa es la base para una cooperación buena y orientada hacia la búsqueda de soluciones. Y es la “base” de una planeación estratégica. Una vez que se define la situación real, analiza los requerimientos específicos.*

## CUESTIONARIO

### De carácter teórico:

1. Defina que es una base de datos.
2. Diferencias entre OLAP y OLTP.
3. Defina modelo de datos.
4. Cuadro comparativo entre los usuarios de base de datos.
5. Establezcan las diferencias entre las etapas para crear una base de datos.

### De investigación:

6. ¿Qué es un Data Warehouse?
7. ¿Qué es Big Data?

### En práctica.

8. Realiza un cuestionario de lo que te gustaría saber de un compañero(o amigo), aplícalo y fundamenta el porqué de cada pregunta realizada.
9. Visita la bodega de tu localidad y realiza una entrevista (grabada) sobre: las ventas, su adquisición de productos y personas con las que trabaja.



## ACTIVIDAD III

### **REPRESENTACION GRAFICA DEL CONTENIDO E INTERACCIÓN DE LA INFORMACIÓN DE LA BASE DE DATOS**

**OBJETIVO:** Realizar un análisis de la información con la que trabaja la empresa y los nuevos requerimientos de esta, estableciendo un modelo conceptual que satisfaga las necesidades de la empresa.

## DISEÑO CONCEPTUAL

En la actividad anterior realizamos la toma de datos de la empresa, en esta actividad aprenderemos a realizar el diseño conceptual de la BD para ello debemos conocer el Modelo De Datos Entidad- Relación.

### Modelo de datos Entidad- Relación

Un **modelo de datos** es una colección de herramientas conceptuales para la descripción de datos, relaciones entre datos, semántica de los datos y restricciones de consistencia.

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos.

Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos.

El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. Debido a esta utilidad, muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R.

Este modelo tiene tres componentes importantes: conjuntos de entidades, conjunto de relaciones y atributos.

## 1. Componentes del Modelo E-R

### 1.1. Conjunto de entidades.

*Una **entidad** es una «cosa» u «objeto» en el mundo real o abstracto que es distinguible de todos los demás objetos.*

Es decir cualquier elemento informativo que tenga importancia para una base de datos.

**Por ejemplo**, cada persona en un desarrollo es una entidad.

Una entidad tiene un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca.

**Por ejemplo**, el D.N.I. 67.789.901 identifica unívocamente una persona particular en la empresa. Análogamente, se puede pensar en los préstamos bancarios como entidades, y un número de préstamo P-15 en la sucursal de cajaP identifica unívocamente una entidad de préstamo.

Una entidad puede ser **concreta**, como una persona o un libro, o



puede ser **abstracta**, como un préstamo, unas vacaciones o un concepto.

Un **conjunto de entidades** es un conjunto de entidades del mismo tipo que comparten las mismas propiedades, o atributos.

El conjunto de todas las personas que son clientes en un banco dado, por ejemplo, se pueden definir como el conjunto de entidades cliente. Análogamente, el **conjunto de entidades** préstamo podría representar el conjunto de todos los préstamos concedidos por un banco particular.

Las entidades individuales que constituyen un conjunto se llaman la extensión del conjunto de entidades. Así, todos los clientes de un banco son la extensión del conjunto de entidades cliente.

Los conjuntos de **entidades** no son necesariamente disjuntos. Por ejemplo, es posible definir el conjunto de entidades de todos los empleados de un banco (empleado) y el conjunto de entidades de todos los clientes del banco (cliente).

Una **entidad** persona puede ser una entidad empleado, una entidad cliente, ambas cosas, o ninguna.

#### **Tipos de entidades:**

- **Regulares.** Son las entidades normales que tienen existencia por sí mismas sin depender de otras.
- **Débiles.** Su existencia depende de otras. **Por ejemplo** la *entidad tarea laboral* sólo podrá tener existencia si existe la entidad trabajo.

### **1.2. Atributos.**

Los **atributos** describen propiedades que posee cada miembro de un conjunto de entidades.

Una entidad se representa mediante un conjunto de atributos.

Los posibles atributos del conjunto de entidades cliente son id-cliente, nombre-cliente, calle-cliente y ciudad-cliente. En la vida real, habría más atributos, tales como el número de la calle, el número del portal, la provincia, el código postal, y la comunidad autónoma, pero no se incluyen en el ejemplo simple. Posibles atributos del conjunto de entidades préstamo son número-préstamo e importe.

Cada entidad tiene un valor para cada uno de sus atributos. Por ejemplo, una entidad cliente en concreto puede tener el valor 10524789 para id-cliente, el valor Santos para nombre-cliente, el valor las Margaritas para callecliente y el valor Chiclayo para ciudad-cliente.

El atributo id-cliente se usa para identificador único de los clientes, dado que puede presentarse más de un cliente con el mismo nombre, calle y ciudad.

También podría utilizar el DNI como idcliente ya que este es único para cada persona o puede optar por la creación de un identificador para cada cliente.

### 1.2.1. Dominio de un atributo.

Es el conjunto de valores permitidos para un atributo, llamados también como el conjunto de valores, de ese atributo.

**Por ejemplo:** El dominio del atributo nombrecliente podría ser el conjunto de todas las cadenas de texto de una cierta longitud.

**Nombrecliente=Gabriel silva Guzmán**->cadena de texto, longitud 20(incluyendo espacios en blanco)

**Análogamente**, el dominio del atributo número-préstamo podría ser el conjunto de todas las cadenas de la forma «P-n», donde n es un entero positivo.

Un **conjunto de entidades** puede tener diferentes atributos, cada entidad se puede describir como un conjunto de pares (atributo,valor), un par para cada atributo del conjunto de entidades.

**Por ejemplo**, una entidad concreta cliente se puede describir mediante el conjunto {(id-cliente, 40268541), (nombre-cliente, López), (calle-cliente, Margaritas), (ciudad-cliente, Chiclayo)}, queriendo decir que la entidad describe una persona llamada López que tiene D.N.I. número 40268541, y reside en la calle margaritas en Chiclayo.

Se puede ver, en este punto, que existe una integración del esquema abstracto con el desarrollo real de la empresa que se está modelando. Los valores de los atributos que describen una entidad constituirán una porción significativa de los datos almacenados en la base de datos.

### 1.2.2. Tipos de atributos.

#### a) Simples:

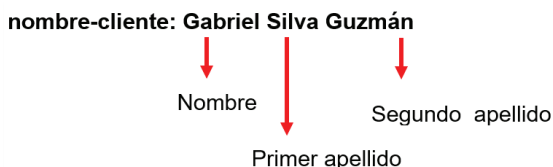
Son aquellos atributos que no se dividen en subpartes.

**Por ejemplo** idcliente si asumimos que es el DNI:40243567 ya no podríamos dividirlo en subpartes.

#### b) Compuestos:

Son aquellos que se pueden dividir en subpartes.

**Por ejemplo**, nombre-cliente podría estar estructurado como un atributo compuesto consistente en nombre, primer-apellido y segundo-apellido.



Usar atributos compuestos en un esquema de diseño es una buena elección si el usuario desea referirse a un atributo completo en algunas ocasiones y, en otro, algún componente del atributo.

Se podrían haber sustituido los atributos del conjunto de entidades cliente calle-cliente y ciudad-cliente, por el atributo compuesto dirección-cliente, con los atributos calle, ciudad, provincia, y código-postal<sup>2</sup>.

Los atributos compuestos ayudan a agrupar los atributos relacionados, haciendo los modelos más claros. Nótese también que un atributo compuesto puede aparecer como una jerarquía.

**c) Monovalorados.**

Son aquellos que tiene valor solo para una entidad.

**Por ejemplo**, el atributo número-préstamo para una entidad préstamo específico, referencia a un único número de préstamo.

**d) Multivalorados.**

Son aquellos atributos que tiene un conjunto de valores; incluso se les puede asignar un límite inferior o superior de valores para una entidad.

**Por ejemplo**, en un conjunto de entidades Cliente el atributo numtelefono; el cual puede tener cero a mas valores.

**e) Derivados.**

El valor para este tipo de atributo se puede derivar de los valores de otros atributos o entidades relacionados.

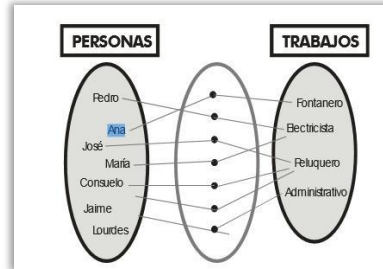
**Ejemplo 1**, sea el conjunto de entidades cliente que tiene un atributo **préstamos** que representa cuántos préstamos tiene un cliente en el banco. Ese atributo se puede derivar contando el número de entidades **préstamo** asociadas con ese **cliente**.

**Ejemplo 2**, considérese que el conjunto de entidades **empleado** tiene un atributo edad, que indica la edad del cliente. Si el conjunto de entidades cliente tiene también un atributo fechade-nacimiento, se **puede calcular edad** a partir de fecha-de-nacimiento y de la fecha actual. Así, edad es un atributo derivado.

### 1.3. Conjunto de relaciones

*Una **relación** es una asociación entre diferentes entidades.*

**Ejemplo 1**, en el caso de que tengamos una entidad personas y otra entidad trabajos. Ambas se realizan ya que las personas trabajan y los trabajos son realizados por personas:



**Ejemplo 2**, se puede definir una relación que asocie al cliente López con el préstamo P-15. Esta relación específica que López es un cliente con el préstamo número P-15.

|            |            |          |            |      |       |
|------------|------------|----------|------------|------|-------|
| 32.112.312 | Santos     | Mayor    | Peguerinos | P-17 | 1.000 |
| 01.928.374 | Gómez      | Carretas | Cercada    | P-23 | 2.000 |
| 67.789.901 | López      | Mayor    | Peguerinos | P-15 | 1.500 |
| 55.555.555 | Sotoca     | Real     | Cádiz      | P-14 | 1.500 |
| 24.466.880 | Pérez      | Carretas | Cercada    | P-19 | 500   |
| 96.396.396 | Valdivieso | Goya     | Vigo       | P-11 | 900   |
| 33.557.799 | Fernández  | Jazmin   | León       | P-16 | 1.300 |

cliente                      préstamo

*¿Qué otras relaciones puedes observar?*




---



---



---

#### 1.3.1. Tipos de relaciones.

- **Relaciones Binarias.** Son las relaciones típicas. Se trata de relaciones que asocian dos entidades.
- **Relaciones Ternarias.** Relacionan tres entidades. A veces se pueden simplificar en relaciones binarias, pero no siempre es posible.
- **Relaciones n-arias.** Relacionan n entidades
- **Relaciones dobles.** Se llaman así a dos relaciones distintas que sirven para relacionar a las mismas relaciones. Son las más difíciles de manejar ya que al manipular las entidades hay que elegir muy bien la relacionan a utilizar para relacionar los datos.
- **Relación reflexiva.** Es una relación que sirve para relacionar ejemplares de la misma entidad (personas con personas, piezas con piezas, etc.)

#### 2. Restricciones.

Un esquema de desarrollo E-R puede definir ciertas restricciones a las que los contenidos de la base de datos se deben adaptar. En este apartado se examina la correspondencia de cardinalidades y las restricciones de participación, que son dos de los tipos más importantes de restricciones.

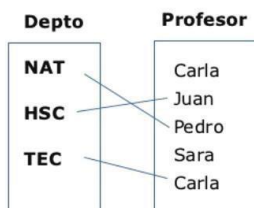
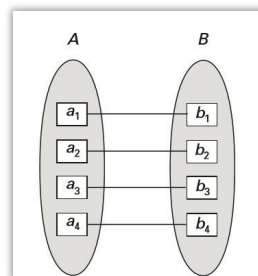
a) **La correspondencia de cardinalidades**

La correspondencia de cardinalidades o razón de cardinalidad, expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

La correspondencia de cardinalidades es la más útil describiendo conjuntos de relaciones binarias (relación que asocia dos entidades). Para un conjunto de relaciones binarias  $R$  entre los conjuntos de entidades  $A$  y  $B$ , la correspondencia de cardinalidades debe ser una de las siguientes:

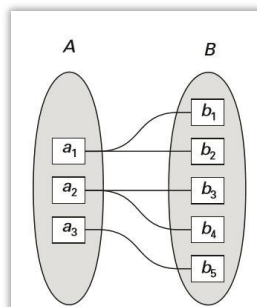
- **Uno a uno.** Una entidad en  $A$  se asocia con a lo sumo una entidad en  $B$ , y una entidad en  $B$  se asocia con a lo sumo una entidad en  $A$ .

Por ejemplo.

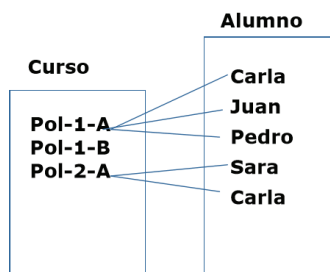


- Todo Departamento tiene solamente un Profesor a cargo
- Un Profesor puede ser jefe de un solo Departamento

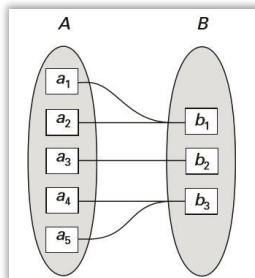
- **Uno a varios.** Una entidad en  $A$  se asocia con cualquier número de entidades en  $B$  (ninguna o varias). Una entidad en  $B$ , sin embargo, se puede asociar con a lo sumo una entidad en  $A$ .



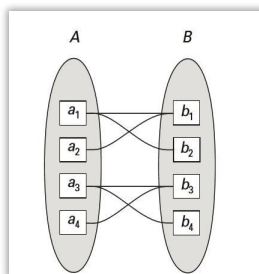
Por ejemplo.



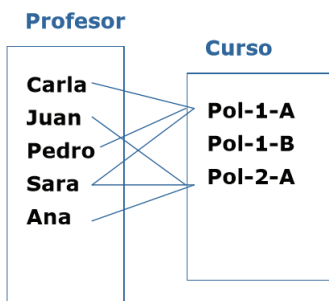
- Un Curso tiene muchos Alumnos
  - Un Alumno está en sólo un curso
  - Un Curso puede estar transitoriamente vacío
  - Todo Alumno tiene un Curso asignado
- **Varios a uno.** Una entidad en A se asocia con a lo sumo una entidad en B. Una entidad en B, sin embargo, se puede asociar con cualquier número de entidades (ninguna o varias) en A.



- **Varios a varios.** Una entidad en A se asocia con cualquier número de entidades (ninguna o varias) en B, y una entidad en B se asocia con cualquier número de entidades (ninguna o varias) en A.



Por ejemplo:



- Un Profesor puede estar asignado en varios Cursos
- Un Curso puede tener muchos Profesores asignados
- Todo Profesor tiene un Curso asignado
- Un Curso puede no tener transitoriamente un Profesor

*La correspondencia de cardinalidades apropiada para un conjunto de relaciones particular depende obviamente de la situación del mundo real que el conjunto de relaciones modela.*

#### Por ejemplo empresa bancaria

Si en un banco particular un préstamo puede pertenecer únicamente a un cliente y un cliente puede tener varios préstamos, **entonces el conjunto de relaciones de cliente a préstamo es uno a varios.**

Si un préstamo puede pertenecer a varios clientes (como préstamos que se toman en conjunto por varios socios de un negocio) **el conjunto de relaciones es varios a varios.**

#### b) Restricciones de participación

La participación de un conjunto de entidades E en un conjunto de relaciones R se dice que es **total** si cada entidad en E participa al menos en una relación en R. Si sólo algunas entidades en E participan en relaciones en R, la participación del conjunto de entidades E en la relación R se llama **parcial**.

**Por ejemplo**, se puede esperar que cada entidad préstamo esté relacionada con al menos un cliente mediante la relación prestatario. Por lo tanto, la participación de préstamo en el conjunto de relaciones prestatario es **total**.

En cambio, un individuo puede ser cliente de un banco tenga o no tenga un préstamo en el banco. Así, es posible que sólo algunas de las entidades cliente estén relacionadas con el conjunto de entidades préstamo mediante la relación prestatario, y la participación de cliente en el conjunto de relaciones prestatario es por lo tanto **parcial**.

### 3. Claves.

Conceptualmente las entidades y relaciones individuales son distintas; desde una perspectiva de bases de datos, sin embargo, la diferencia entre ellas se debe expresar en término de sus atributos. Por lo tanto, los valores de los atributos de una entidad deben ser tales que permitan identificar unívocamente la entidad.

*Una clave permite identificar un conjunto de atributos suficiente para distinguir las entidades entre sí. Las claves también ayudan a identificar unívocamente a las relaciones y así a distinguir las relaciones entre sí.*

#### 3.1. Superclave.

Es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades.

**Por ejemplo**, el atributo id-cliente del conjunto de entidades cliente es suficiente para distinguir una entidad cliente de las otras. Así, id-cliente es una superclave. Análogamente, la combinación de nombre-cliente e idcliente es una superclave del conjunto de **entidades cliente**. El atributo nombre-cliente de cliente no es una superclave, porque varias personas podrían tener el mismo nombre.

#### 3.2. Clave primaria.

Se utiliza para denotar una clave candidata que es elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades. Una clave (primaria, candidata y superclave) es una propiedad del conjunto de entidades, más que de las entidades individuales. Cualesquiera dos entidades individuales en el conjunto no pueden tener el mismo valor en sus atributos clave al mismo tiempo. La designación de una clave representa una restricción en el **desarrollo del mundo real** que se modela.

**La clave primaria** se debería elegir de manera que sus atributos nunca, o muy raramente, cambien

#### 3.3. Clave candidata.

Atributos que cumplen las condiciones de identificación única de entidades, pero que no fueron definidos como principales por el diseñador.

Las claves candidatas se deben designar con cuidado. Como se puede comprender, el nombre de una persona es obviamente insuficiente, ya que hay mucha gente con el mismo nombre. Una alternativa es usar alguna combinación única de otros atributos como clave.

#### 4. Diagrama Entidad – Relación.

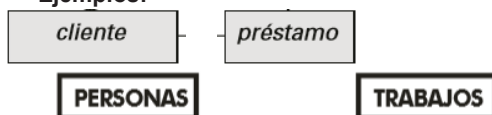
Un diagrama entidad-relación (ER) es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema.

También conocidos como los ERD o modelos ER, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos. Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos.

##### 4.1. Elementos gráficos:

- **Rectángulos**, que representan conjuntos de entidades.

Ejemplos:



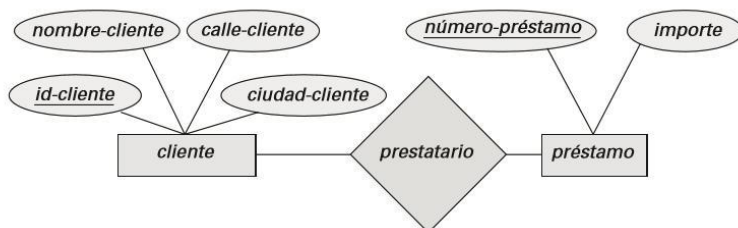
- **Elipses**, que representan atributos.

Ejemplos



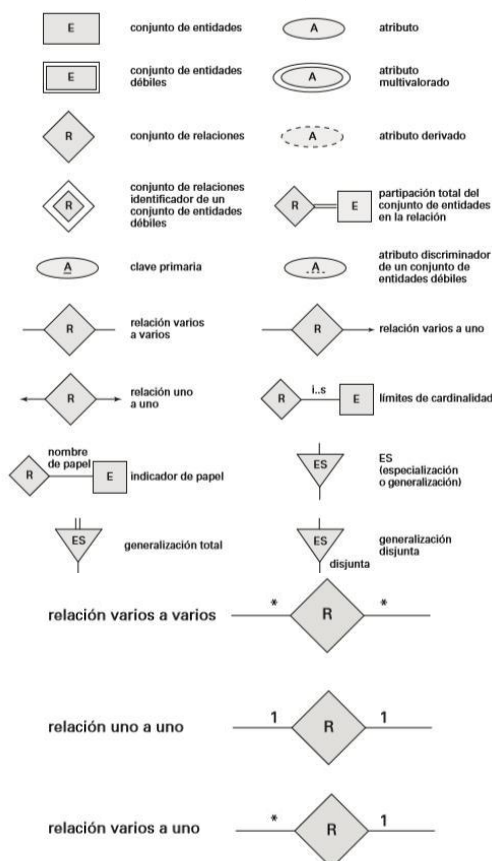
- **Rombos**, que representan relaciones.

Ejemplos:



- **Líneas**, que unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones.
- **Elipses dobles**, que representan atributos multivalorados.
- **Elipses discontinuas**, que denotan atributos derivados.

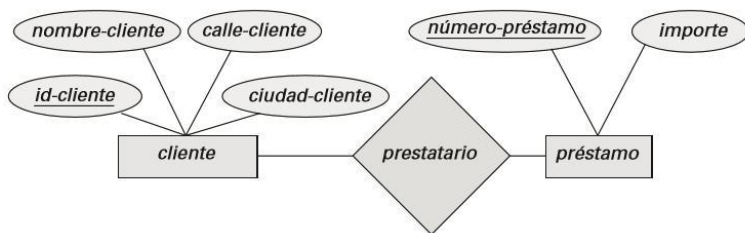
- **Líneas dobles**, que indican participación total de una entidad en un conjunto de relaciones.
- **Rectángulos dobles**, que representan conjuntos de entidades débiles.



#### 4.2. Complementación

Considérese el diagrama entidad-relación que consta de dos conjuntos de entidades, cliente y préstamo, relacionadas a través de un conjunto de relaciones binarias prestatario.

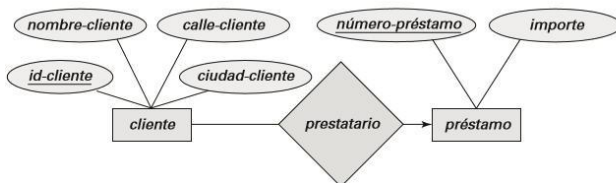
- Los atributos asociados con cliente son id-cliente, nombre-cliente, calle-cliente, y ciudad-cliente. Los atributos asociados con préstamo son número-préstamo e importe.
- Los atributos de un conjunto de entidades que son miembros de la clave primaria están subrayados.
- El conjunto de relaciones prestatario puede ser varios a varios, uno a varios, varios a uno o uno a uno.
- Para distinguir entre estos tipos, se dibuja o una línea dirigida ( $\rightarrow$ ) o una línea no dirigida ( $\text{—}$ ) entre el conjunto de relaciones y el conjunto de entidades en cuestión.



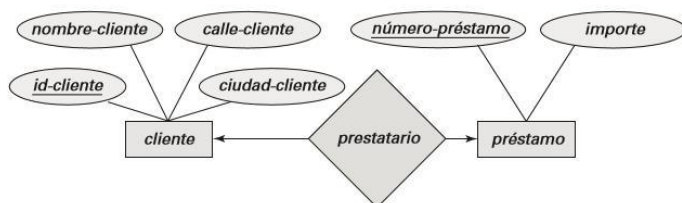
- Una línea dirigida desde el conjunto de relaciones prestatario al conjunto de entidades préstamo especifica que prestatario es un conjunto de relaciones uno a uno, o bien varios a uno, desde cliente a préstamo; prestatario no puede ser un conjunto de relaciones varios a varios, desde cliente a préstamo.
- **Una línea no dirigida desde** el conjunto de relaciones prestatario al conjunto de relaciones préstamo especifica que prestatario es o bien un conjunto de relaciones varios a varios, o bien uno a varios, desde cliente a préstamo.
- Volviendo **al diagrama E-R** se ve que el conjunto de relaciones prestatario es varios a varios. Si el conjunto de relaciones prestatario fuera uno a varios, desde cliente a préstamo, entonces la línea desde prestatario a cliente sería dirigida, con una flecha apuntando al conjunto de entidades cliente.



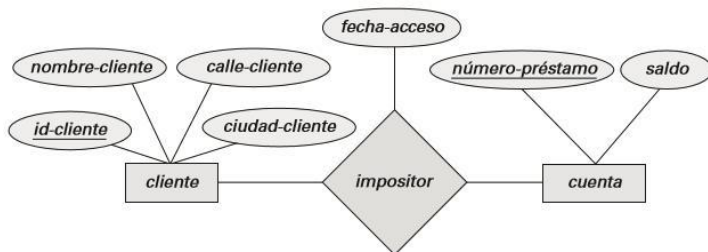
- **Análogamente, si el conjunto** de relaciones prestatario fuera varios a uno desde cliente a préstamo, entonces la línea desde prestatario a préstamo tendría una flecha apuntando al conjunto de entidades préstamo.



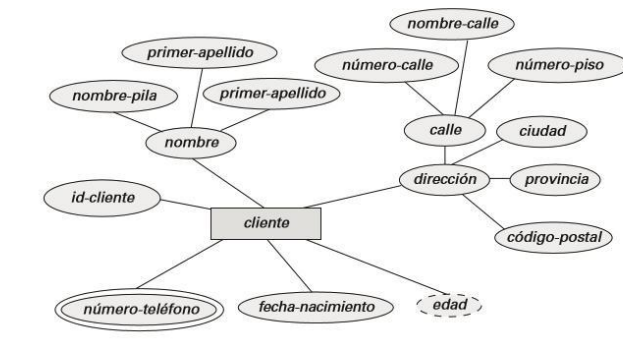
- **Finalmente, si el conjunto** de relaciones prestatario fuera uno a uno, entonces ambas líneas desde prestatario tendrían flechas: una apuntando al conjunto de entidades préstamo y otra apuntando al conjunto de entidades cliente.



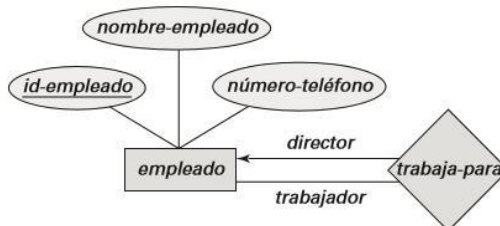
- Si un conjunto de relaciones tiene también algunos atributos asociados a él, entonces se unen esos atributos a ese conjunto de relaciones. **Por ejemplo**, se tiene el atributo descriptivo **fecha-acceso** unida al conjunto de relaciones impositor para especificar la fecha más reciente en la que un cliente accedió a esa cuenta.



- En la imagen se muestra cómo se pueden representar atributos compuestos en la notación E-R. Aquí, el atributo **compuesto nombre**, con atributos componentes **nombre-pila**, **primer-apellido** y **segundo-apellido** reemplaza al atributo simple **nombre-cliente** de **cliente**. También se muestra **el atributo compuesto** dirección, cuyos atributos componentes son calle, ciudad, provincia y código-postal, que reemplaza a los atributos **calle-cliente** y **ciudad-cliente** de **cliente**. El atributo **calle** es por sí mismo un atributo compuesto cuyos atributos componentes son número-calle, nombre-calle y número-piso. También muestra un atributo **multivariado**, **número-teléfono**, indicado por una elipse doble, y un atributo derivado edad, indicado por una elipse discontinua.



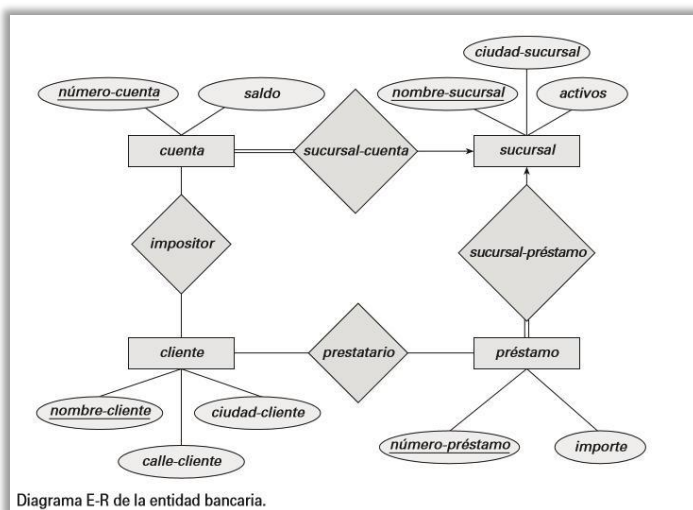
- En los diagramas E-R se indican papeles mediante etiquetas en las líneas que unen rombos con rectángulos. En la Figura 2.12 se muestran los indicadores de papeles **director** y **trabajador** entre el conjunto de entidades empleado y el conjunto de relaciones trabaja para.



- Los conjuntos de relaciones **no binarias** se pueden especificar fácilmente en un diagrama E-R. La Figura 2.13 consta de tres conjuntos de entidades **cliente**, **trabajo** y **sucursal**, relacionados a través del conjunto de relaciones **trabaja-en**.

Se pueden especificar algunos tipos de relaciones varios a uno en el caso de conjuntos de relaciones no binarias. Supóngase un empleado que tenga a lo sumo un trabajo en cada sucursal (por ejemplo, Santos no puede ser director y auditor en la misma sucursal). Esta restricción se puede especificar con una flecha apuntando a trabajo en el borde de **trabaja-en**.





## CUESTIONARIO

### Teórico:

1. Defina y represente gráficamente una entidad y atributo.
2. ¿Cuáles son los tipos de atributos que podemos encontrar en una entidad, de ejemplos?
3. ¿Qué es una relación?
4. Explíquense las diferencias entre los términos clave primaria, clave candidata y superclave.

### Practico:

1. Representa gráficamente las entidades encontradas en la información recolectada en la Actividad I(bodega)
2. Crear un diccionario de datos:
  - a. Indica el nombre de cada entidad e indica su función.
  - b. Indica los atributos de cada entidad e indica que valor va almacenar.
3. Realiza el Diagrama E-R del proyecto Bodega.

### Investigación:

1. Explíquense las diferencias entre conjunto de entidades débiles y fuertes
2. Realizar un organizador grafico sobre Sistema de gestión de base de datos.
3. ¿Qué usos se le pueden dar a un diagrama entidad- relación?



# ACTIVIDAD IV

## ESQUEMA RELACIONAL DE LA BASE DE DATOS DE LA EMPRESA

**OBJETIVO:** Transformar el esquema conceptual obtenido en la etapa anterior, adaptándolo a un conjunto de estructuras propias del modelo de datos elegido; modelo relacional.

## DISEÑO LÓGICO

### 1. Modelo Relacional

El modelo relacional se ha establecido actualmente como el principal modelo de datos para las aplicaciones de procesamiento de datos. Ha conseguido la posición principal debido a su simplicidad, que facilita el trabajo.

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el **Álgebra relacional** y el **Cálculo relacional**.

#### 1.1. Objetivos:

- **Independencia física.**

La forma de almacenar los datos, no debe influir en su manipulación lógica. Si la forma de almacenar los datos cambia, los usuarios no tienen siquiera porque percibirlo y seguirán trabajando de la misma forma con la base de datos. Esto permite que los usuarios y usuarias se concentren en qué quieren consultar en la base de datos y no en cómo está realizada la misma.

- **Independencia lógica.** Las aplicaciones que utilizan la base de datos no deben ser modificadas porque se modifiquen elementos de la base de datos. Es decir, añadir, borrar y suprimir datos, no influye en las vistas de los usuarios. De una manera más precisa, gracias a esta independencia el esquema externo de la base de datos es realmente independiente del modelo lógico.

- **Flexibilidad.** La base de datos ofrece fácilmente distintas vistas en función de los usuarios y aplicaciones. Uniformidad. Las estructuras lógicas siempre tienen una única forma conceptual (las relaciones).

- **Sencillez.** Facilidad de manejo (algo cuestionable, pero ciertamente verdadero si comparamos con los sistemas gestores de bases de datos anteriores a este modelo).

#### 1.2. Estructura.

##### 1.2.1. Relaciones o relaciones.

Es el elemento principal del modelo relacional, ya que una base de datos relacional está formada por un conjunto de relaciones o relaciones.

Codd definió las relaciones utilizando un lenguaje matemático, pero se pueden asociar a la idea de relación (de filas y columnas) ya que es más fácil de entender. **No hay que confundir la idea de relación según el modelo de Codd, con lo que significa una relación en el modelo Entidad/Relación de Chen.**

a) Las relaciones constan de:

- **Atributos.** Referido a cada propiedad de los datos que se almacenan en la relación (número-cuenta, nombre-sucursal, saldo). Para cada atributo hay un conjunto de valores permitidos, llamado dominio de ese atributo.

**Por ejemplo:** Para el atributo nombre-sucursal, el dominio es el conjunto de los nombres de las sucursales.

- **Tuplas.** Referido a cada elemento de la relación. Por ejemplo si una **relación** (relación) almacena cuentas, una tupla representaría a una cuenta en concreto. Puesto que una relación se representa como una relación; podemos entender que las columnas de la relación son los atributos; y las filas, las tuplas.

| número-cuenta | nombre-sucursal | saldo |
|---------------|-----------------|-------|
| C-101         | Centro          | 500   |
| C-102         | Navacerrada     | 400   |
| C-201         | Galapagar       | 900   |
| C-215         | Becerril        | 700   |
| C-217         | Galapagar       | 750   |
| C-222         | Moralzarzal     | 700   |
| C-305         | Collado Mediano | 350   |

- **Cuerpo de la relación.** Representa el conjunto de  $m$  tuplas  $\{t_1, t_2, \dots, t_n\}$  que forman la relación. Cada tupla es un conjunto de  $n$  pares atributo-valor  $\{(A_i, V_{ij})\}$ , donde  $V_{ij}$  es el valor  $j$  del dominio  $D_i$  asociado al atributo  $A_i$ .
- **Esquema de la relación.** Se adopta el convenio de utilizar nombres en minúsculas para las relaciones y nombres que comiencen por una letra mayúscula para los esquemas de las relaciones.

Siguiendo esta notación se utilizará **Esquema-cuenta** para denotar el **esquema de la relación de la relación cuenta**. Por tanto,

Esquema-cuenta = (número-cuenta, nombre-sucursal, saldo)

Se denota el hecho de que cuenta es una relación de Esquema-cuenta mediante

cuenta (Esquema-cuenta)

En general, los esquemas de las relaciones incluyen una lista de los atributos y de sus dominios correspondientes.

b) **Propiedades de las relaciones (o relaciones)**

- Cada relación tiene un nombre distinto
- Cada atributo de la relación toma un solo valor en cada tupla
- Cada atributo tiene un nombre distinto en cada relación (aunque puede coincidir en relaciones distintas)
- Cada tupla es única (no hay tuplas duplicadas)
- El orden de los atributos no importa.
- El orden de las tuplas no importa.

1.2.2. **Claves**

Los conceptos **de superclave, de clave candidata y de clave primaria**, tal y como se discute en la actividad anterior, también son aplicables en el **modelo relacional**.

Por ejemplo, en Esquema-sucursal, tanto {nombre-sucursal} como {nombre-sucursal, ciudad-sucursal} son superclaves. {nombre-sucursal, ciudad-sucursal} no es una clave candidata porque {nombre-sucursal} es un subconjunto de {nombre-sucursal, ciudad-sucursal} y {nombre-sucursal} es una superclave. Sin embargo, {nombre-sucursal} es una clave candidata, y servirá también como clave primaria para estos fines. El atributo **ciudad-sucursal** no es una superclave, dado que dos sucursales de la misma ciudad pueden tener nombres diferentes (y diferentes volúmenes de activos).

Si el esquema de una base de datos relacional se basa en las relaciones derivadas de un **esquema E-R** es posible determinar la clave primaria del esquema de una relación a partir de las claves primarias de los conjuntos de entidades o de relaciones de los que se deriva el esquema:

- **Conjunto de entidades fuertes.** La clave primaria del conjunto de entidades se convierte en la clave primaria de la relación.
- **Conjunto de entidades débiles.** La relación y, por tanto, la relación correspondientes a un conjunto de entidades débiles incluyen:
  - Los atributos del conjunto de entidades débiles.
  - La clave primaria del conjunto de entidades fuertes del que depende el conjunto de entidades débiles.

La clave primaria de la relación consiste en la unión de la clave primaria del conjunto de entidades fuertes y el discriminante del conjunto de entidades débil.

- **Conjunto de relaciones.** La unión de las claves primarias de los conjuntos de entidades relacionados se transforma en una superclave de la relación. Si la relación es de varios a varios, esta superclave es también la clave primaria.
- **Relaciones combinadas.** Recuerdese que un conjunto binario de relaciones de **varios a uno** entre A y B puede representarse mediante una relación que consista en los atributos de A y en los atributos (si hay alguno) del conjunto de relaciones. La clave primaria de la entidad

«varios» se transforma en la clave primaria de la relación (es decir, si el conjunto de relaciones es de varios a uno entre A y B, la clave primaria de A es la clave primaria de la relación). Para los conjuntos de relaciones de uno a uno la relación se construye igual que en el conjunto de relaciones de varios a uno. Sin embargo, cualquiera de las claves primarias del conjunto de entidades puede elegirse como clave primaria de la relación, dado que ambas son claves candidatas.

- **Atributos multivalorados.** Recuérdese que un atributo multivalorado **M** se representa mediante una relación consistente en la clave primaria del conjunto de entidades o de relaciones del que **M** es atributo y en una columna **C** que guarda un valor concreto de **M**. La clave primaria del conjunto de entidades o de relaciones junto con el atributo **C** se convierte en la clave primaria de la relación.
- **Clave externa, ajena o secundaria.** Son los datos de atributos de una relación cuyos valores están relacionados con atributos de otra relación. Por ejemplo en la relación equipos tenemos estos datos:

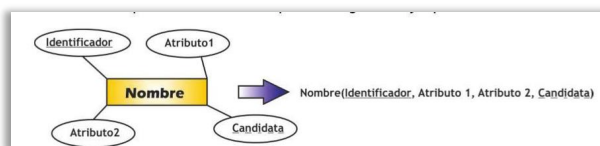
### 1.3. Paso de entidad/relación de Chen al modelo relacional

#### 1.3.1. Transformación de las entidades fuertes:

En principio las entidades fuertes del modelo Entidad Relación son transformadas al modelo relacional siguiendo estas instrucciones:

- **Entidades.** Las entidades pasan a ser relaciones.
- **Atributos.** Los atributos pasan a ser columnas o atributos de la relación.
- **Identificadores principales.** Pasan a ser claves primarias.
- **Identificadores candidatos.** Pasan a ser claves candidatas.

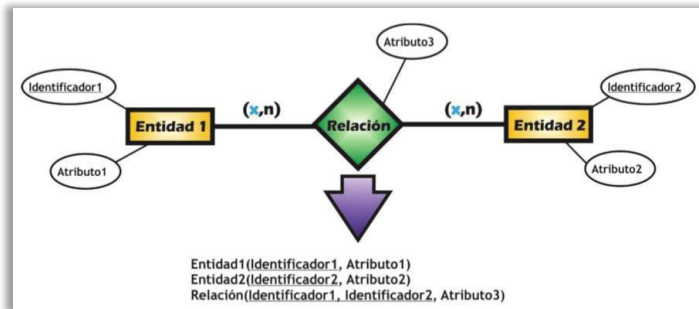
Esto hace que la transformación se produzca según este ejemplo:



#### 1.3.2. Transformación de relaciones

- **Relaciones varios a varios.**

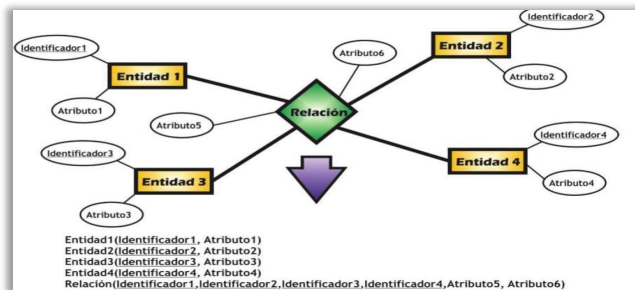
En las relaciones varios a varios (**n a n** en la cardinalidad mayor, la cardinalidad menor no cuenta para esta situación), la relación se transforma en una relación cuyos atributos son: los atributos de la relación y las claves de las entidades relacionadas (que pasarán a ser claves externas). La clave de la relación la forman todas las claves externas.



En el dibujo anterior los símbolos x referidos a la cardinalidad mínima, indican que ésta no influye en el paso al modelo relacional.

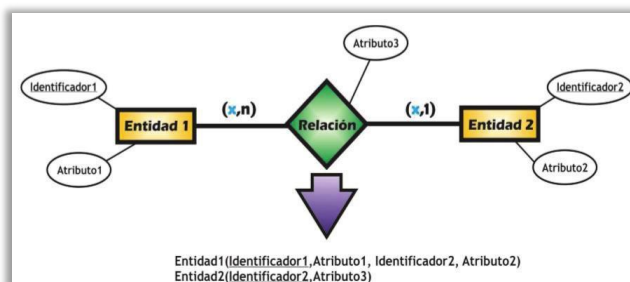
- **Relaciones de orden n.**

Las relaciones ternarias, cuaternarias y n-arias que unen más de dos relaciones se transforman en una relación que contiene los atributos de la relación más los identificadores de las entidades relacionadas. La clave la forman todas las claves externas:



- **Relaciones uno a varios.**

Las relaciones binarios de tipo uno a varios no requieren ser transformadas en una relación en el modelo relacional. En su lugar la relación del lado varios (relación relacionada) incluye como clave externa1 el identificador de la entidad del lado uno (relación principal).

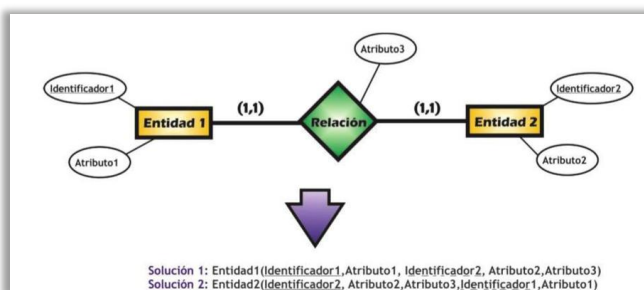


Así en el dibujo, el Identificador2 en la relación Entidad1 pasa a ser una clave secundaria. En el caso de que el número mínimo de la relación sea de cero (puede haber ejemplares de la entidad uno sin relacionar), se deberá permitir valores nulos en la clave secundaria (en el ejemplo sería el identificador2 en la Entidad1). En otro caso no se podrán permitir (ya que siempre habrá un valor relacionado).

- **Relaciones uno a uno**

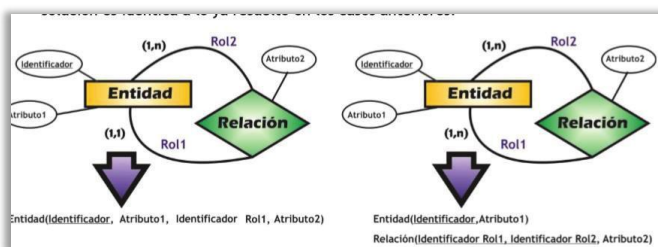
En el caso de las relaciones entre dos entidades con todas las cardinalidades a 1; hay dos posibilidades:

- **Colocar la clave de una de las entidades** como clave externa de la otra relación (da igual cuál), teniendo en cuenta que dicha clave será clave alternativa además de ser clave secundaria.
- **Generar una única relación** con todos los atributos de ambas entidades colocando como clave principal cualquiera de las claves de las dos entidades. La otra clave será marcada como clave alternativa. El nombre de la relación sería el de la entidad más importante desde el punto de vista conceptual.

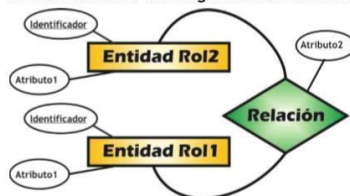


- **Relaciones recursivas**

Las relaciones recursivas se tratan de la misma forma que las otras, sólo que hay que imaginar que la relación se divide en dos, una por cada rol. Teniendo en cuenta eso, la solución es idéntica a lo ya resuelto en los casos anteriores.



Es decir la forma de imaginarse las relaciones recursivas para resolverlas es:



#### 1.4. Ejemplo empresa bancaria

Relación: Sucursal

| nombre de la sucursal | ciudad de la sucursal | activos   |
|-----------------------|-----------------------|-----------|
| Galapagar             | Arganzuela            | 7.500     |
| Centro                | Arganzuela            | 9.000.000 |
| Becerril              | Aluche                | 2.000     |
| Segovia               | Cerceda               | 3.700.000 |
| Navacerrada           | Aluche                | 1.700.000 |
| Navas de la Asunción  | Alcalá de Henares     | 1.500     |
| Moralzarzal           | La Granja             | 2.500     |
| Collado Mediano       | Aluche                | 8.000.000 |

Relación: Cliente

| nombre-cliente | calle-cliente | ciudad-cliente |
|----------------|---------------|----------------|
| Abril          | Preciados     | Valsain        |
| Amo            | Embajadores   | Arganzuela     |
| Badorrey       | Delicias      | Valsain        |
| Fernández      | Jazmín        | León           |
| Gómez          | Carretas      | Cerceda        |
| González       | Arenal        | La Granja      |
| López          | Mayor         | Peguerinos     |
| Pérez          | Carretas      | Cerceda        |
| Rodríguez      | Yaserías      | Cádiz          |
| Rupérez        | Ramblas       | León           |
| Santos         | Mayor         | Peguerinos     |
| Valdivieso     | Goya          | Vigo           |

Relación: Impositor

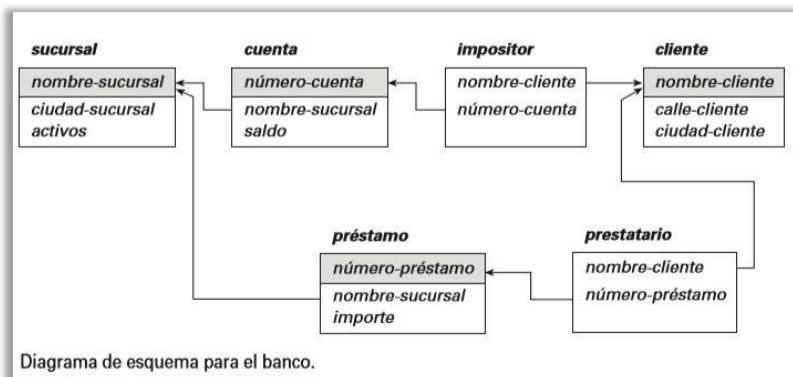
| nombre cliente | número cuenta |
|----------------|---------------|
| Abril          | C-102         |
| Gómez          | C-101         |
| González       | C-201         |
| González       | C-217         |
| López          | C-222         |
| Rupérez        | C-215         |
| Santos         | C-305         |

Relación: Préstamo

| número-préstamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-11            | Collado Mediano | 900     |
| P-14            | Centro          | 1.500   |
| P-15            | Navacerrada     | 1.500   |
| P-16            | Navacerrada     | 1.300   |
| P-17            | Centro          | 1.000   |
| P-23            | Moralzarzal     | 2.000   |
| P-93            | Becerril        | 500     |

Relación: Prestario

| nombre cliente | número préstamo |
|----------------|-----------------|
| Fernández      | P-16            |
| Gómez          | P-93            |
| Gómez          | P-15            |
| López          | P-14            |
| Pérez          | P-17            |
| Santos         | P-11            |
| Sotoca         | P-23            |
| Valdivieso     | P-17            |



## CUESTIONARIO

**Teórico:**

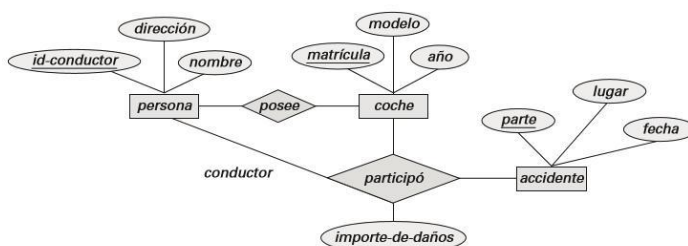
1. Defina modelo relacional y principales ventajas.
2. Defina relación y dar tres ejemplos.
3. Cree una relación, sus atributos y clave primaria.
4. Realice cuadro comparativo modelo E-R y modelo relacional.

**Investigación:**

5. Realizar un organizador visual que represente la historia del Modelo Relacional.
6. Herramientas tecnológicas para diagramar el E-R y modelo Relacional.

**Practico:**

7. Diséñese una base de datos relacional correspondiente al diagrama E-R de la imagen.



8. Del proyecto bodega diseñe la base de datos relacional.

# ACTIVIDAD V

## **RESTRICCIONES EN EL DISEÑO DE LA BASE DE DATOS RELACIONAL DE LA EMPRESA BASADOS EN LA NORMALIZACION DE DATOS**

OBJETIVO: Resolver los posibles problemas de diseño de la base de datos relacional utilizando las formas normales.

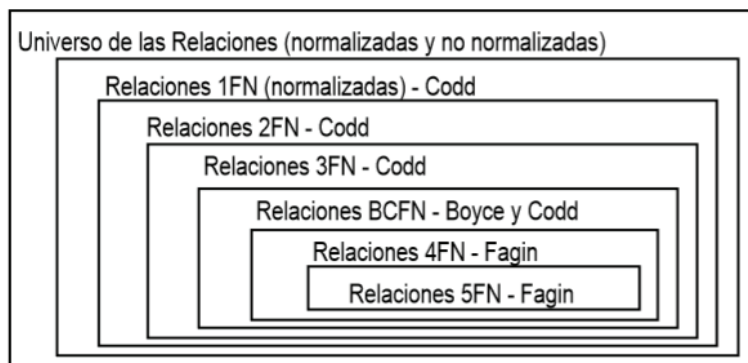
## NORMALIZACIÓN

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo a arriba: se parte de los atributos y éstos se van agrupando en relaciones (relacions) según su afinidad. Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.

En el proceso de normalización se debe ir comprobando que cada relación (relación) cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

La normalización se lleva a cabo en una serie de pasos. Cada paso corresponde a una forma normal que tiene unas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto (más fuerte) y, por lo tanto, son menos vulnerables a las anomalías de actualización. El modelo relacional sólo requiere un conjunto de relaciones en primera forma normal. Las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.



### 1. Problemas Del Esquema Relacional

Una vez obtenido el esquema relacional resultante del esquema entidad/relación que representa la base de datos, normalmente tendremos una buena base de datos. Pero otras veces, debido a fallos en el diseño o a problemas indetectables, tendremos un esquema que puede producir una base de datos que incorpore estos problemas:

- **Redundancia.** Se llama así a los datos que se repiten continua e innecesariamente por las relaciones de las bases de datos. Cuando es excesiva es evidente que el diseño hay que revisarlo, es el primer síntoma de problemas y se detecta fácilmente.
- **Ambigüedades.** Datos que no clarifican suficientemente el elemento al que representan. Los datos de cada fila podrían referirse a más de un ejemplar de esa relación o incluso puede ser imposible saber a qué ejemplar exactamente se están refiriendo. Es un problema muy grave y difícil de detectar.

- **Pérdida de restricciones de integridad.** Normalmente debido a dependencias funcionales. Más adelante se explica este problema. Se arreglan fácilmente siguiendo una serie de pasos concretos.
- **Anomalías en operaciones de modificación de datos.** El hecho de que al insertar un solo elemento haya que repetir tuplas en una relación para variar unos pocos datos. O que eliminar un elemento suponga eliminar varias tuplas necesariamente (por ejemplo que eliminar un cliente suponga borrar seis o siete filas de la relación de clientes, sería un error muy grave y por lo tanto un diseño terrible).

## 2. Primera forma normal (1FN)

Se dice que el esquema de una relación R está en la primera forma normal (1FN) si los dominios de todos los atributos de R son atómicos.

Un **dominio es atómico** si se considera que los elementos del dominio son unidades indivisibles.

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos.

**Por ejemplo 1**, si el esquema de la relación **empleado** incluyera el **atributo hijos**, los elementos de cuyo dominio son conjuntos de nombres, el esquema no se hallaría en la primera forma normal.

| Nombre-empleado | Edad-empleado | Hijos-empleado    |
|-----------------|---------------|-------------------|
| juan            | 25            | Mario, Juan y Ana |

**Ejemplo 2:** La Relación: TRABAJADOR

| TRABAJADOR |        |                      |
|------------|--------|----------------------|
| DNI        | Nombre | Departamento         |
| 12121212A  | Andrés | Mantenimiento        |
| 12345345G  | Andrea | Dirección<br>Gestión |

Visualmente es una relación, pero **NO** cumple la primera forma normal.

## 3. Dependencias funcionales.

La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos.

### 3.1. Dependencia funcional.

Se dice que un conjunto de atributos (Y) depende funcionalmente de otro conjunto de atributos (X) si para cada valor de X hay un único valor posible para Y.

*Simbólicamente se denota por  $X \rightarrow Y$ .*

**Por ejemplo** el nombre de una persona depende funcionalmente del DNI; es decir para un DNI concreto sólo hay un nombre posible. En la relación del ejemplo anterior, el departamento no tiene dependencia funcional, ya que para un mismo DNI puede haber más de un departamento posible. Pero el nombre sí que depende del DNI. Al conjunto X del que depende funcionalmente el conjunto Y se le llama determinante. Al conjunto Y se le llama implicado.

**3.2. Dependencia funcional completa** Un conjunto de atributos (Y) tiene una dependencia funcional completa sobre otro conjunto de atributos (X) si Y tiene dependencia funcional de X y además no se puede obtener de X un conjunto de atributos más pequeño que consiga una dependencia funcional de Y (es decir, no hay en X un determinante formado por atributos más pequeños).

**Por ejemplo** en una relación de clientes, el conjunto de atributos formado por el nombre y el DNI producen una dependencia funcional sobre el atributo **apellidos**. Pero no es plena ya que el **DNI** individualmente, también produce una dependencia funcional sobre apellidos. El **DNI** sí produce una dependencia funcional completa sobre el campo apellidos.

*Una dependencia funcional completa se denota como  $X \twoheadrightarrow Y$*

**3.3. Dependencia funcional elemental.** Se produce cuando X e Y forman una dependencia funcional completa y además Y es un único atributo.

**3.4. Dependencia funcional transitiva** Es más compleja de explicar, pero tiene también utilidad. Se produce cuando tenemos tres conjuntos de atributos X, Y y Z. Y depende funcionalmente de X ( $X \rightarrow Y$ ), Z depende funcionalmente de Y ( $Y \rightarrow Z$ ). Además X no depende funcionalmente de Y ( $Y \not\rightarrow X$ ). Entonces ocurre que X produce una dependencia funcional transitiva sobre Z. Esto se denota como: ( $X \twoheadrightarrow Z$ ) Por ejemplo si X es el atributo Número de Clase de un instituto, e Y es el atributo Código Tutor. Entonces  $X \rightarrow Y$  (el tutor depende funcionalmente del número de clase). Si Z representa el Código del departamento, entonces  $Y \rightarrow Z$  (el código del departamento depende funcionalmente del código tutor, cada tutor sólo puede estar en un departamento). Como ocurre que  $Y \not\rightarrow X$  (el código de la clase no depende funcionalmente del código tutor, un código tutor se puede corresponder con varios códigos de clase). Entonces  $X \twoheadrightarrow Z$  (el código del departamento depende transitivamente del código de la clase).

## 4. Segunda forma normal (2FN)

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).

| ALUMNOS    |                  |        |           |      |
|------------|------------------|--------|-----------|------|
| <u>DNI</u> | <u>Cod Curso</u> | Nombre | Apellido1 | Nota |
| 12121219A  | 34               | Pedro  | Valiente  | 9    |
| 12121219A  | 25               | Pedro  | Valiente  | 8    |
| 3457775G   | 34               | Ana    | Fernández | 6    |
| 5674378J   | 25               | Sara   | Crespo    | 7    |
| 5674378J   | 34               | Sara   | Crespo    | 6    |

Suponiendo que el DNI y el código de curso formen una clave principal para esta relación, sólo la nota tiene dependencia funcional completa. El nombre y los apellidos dependen de forma completa del DNI. La relación no es 2FN, para arreglarlo:

| ALUMNOS    |        |           |
|------------|--------|-----------|
| <u>DNI</u> | Nombre | Apellido1 |
| 12121219A  | Pedro  | Valiente  |
| 3457775G   | Ana    | Fernández |
| 5674378J   | Sara   | Crespo    |

| ASISTENCIA |                  |      |
|------------|------------------|------|
| <u>DNI</u> | <u>Cod Curso</u> | Nota |
| 12121219A  | 34               | 9    |
| 12121219A  | 25               | 8    |
| 3457775G   | 34               | 6    |
| 5674378J   | 25               | 7    |
| 5674378J   | 34               | 6    |

### 5. Tercera forma normal (3FN)

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. La dependencia es transitiva si existen las dependencias , , siendo , , atributos o conjuntos de atributos de una misma relación.

Aunque las relaciones en 2FN tienen menos redundancias que las relaciones en 1FN, todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que dependen).

| ALUMNOS   |          |           |               |            |
|-----------|----------|-----------|---------------|------------|
| DNI       | Nombre   | Apellido1 | Cod Provincia | Provincia  |
| 12121349A | Salvador | Velasco   | 34            | Palencia   |
| 12121219A | Pedro    | Valiente  | 34            | Palencia   |
| 3457775G  | Ana      | Fernández | 47            | Valladolid |
| 5674378J  | Sara     | Crespo    | 47            | Valladolid |
| 3456858S  | Marina   | Serrat    | 08            | Barcelona  |

La Provincia depende funcionalmente del código de provincia, lo que hace que no esté en 3FN. El arreglo sería:

| ALUMNOS   |          |           |               |
|-----------|----------|-----------|---------------|
| DNI       | Nombre   | Apellido1 | Cod Provincia |
| 12121349A | Salvador | Velasco   | 34            |
| 12121219A | Pedro    | Valiente  | 34            |
| 3457775G  | Ana      | Fernández | 47            |
| 5674378J  | Sara     | Crespo    | 47            |
| 3456858S  | Marina   | Serrat    | 08            |

| PROVINCIA     |            |
|---------------|------------|
| Cod Provincia | Provincia  |
| 34            | Palencia   |
| 47            | Valladolid |
| 08            | Barcelona  |

## 6. Forma normal de Boyce-Codd (FNBC o BCFN)

Esta forma normal se definió para resolver las anomalías de actualización provocadas por un caso muy específico, que se da cuando:

- La relación tiene varias claves candidatas.
- Las claves candidatas están solapadas en algún subconjunto de atributos de las dos. Si cualquiera de estas dos condiciones no se cumpliera, normalizar hasta 3FN sería suficiente.

**Una relación está en forma normal de Boyce-Codd (FNBC) si y sólo si todo determinante es una clave candidata.**

Definimos **determinante** como un conjunto de atributos del que depende funcionalmente por completo algún otro atributo.

Ejemplo:

| ORGANIZACIÓN |              |             |
|--------------|--------------|-------------|
| Trabajador   | Departamento | Responsable |
| Alex         | Producción   | Felipa      |
| Arturo       | Producción   | Martín      |
| Carlos       | Ventas       | Julio       |
| Carlos       | Producción   | Felipa      |
| Gabriela     | Producción   | Higinio     |
| Luisa        | Ventas       | Eva         |
| Luisa        | Producción   | Martín      |
| Manuela      | Ventas       | Julio       |
| Pedro        | Ventas       | Eva         |

Un trabajador o trabajadora puede trabajar en varios departamentos.

En dicho departamento hay varios responsables, pero cada trabajador sólo tiene asignado uno. El detalle importante que no se ha tenido en cuenta, es que el o la responsable sólo puede ser responsable en un departamento.

Este detalle último produce una dependencia funcional ya que:

**Responsable → Departamento**

**Por lo tanto** hemos encontrado un determinante que no es clave candidata. No está por tanto en FNBC. En este caso la redundancia ocurre por mala selección de clave. La redundancia del departamento es completamente evitable. La solución sería:

| PERSONAL   |             |
|------------|-------------|
| Trabajador | Responsable |
| Alex       | Felipa      |
| Arturo     | Martín      |
| Carlos     | Julio       |
| Carlos     | Felipa      |
| Gabriela   | Higinio     |
| Luisa      | Eva         |
| Luisa      | Martín      |
| Manuela    | Julio       |
| Pedro      | Eva         |

## 7. Cuarta forma normal (4FN). Dependencias Multivaluadas

- **Dependencia multivaluada**

Si las dependencias funcionales eran la base de la segunda y tercera forma normal (y de la de Boyce-Codd), éstas son la base de la cuarta forma normal. Una dependencia multivaluada de X sobre Y (es decir  $X \twoheadrightarrow Y$ ), siendo X e Y atributos de la misma relación, ocurre cuando Y tiene un conjunto de valores bien definidos sobre cualquier valor de X. Es decir, dado X sabremos los posibles valores que puede tomar Y. Se refiere a posibles valores (en plural) y se trata de que los valores de ese atributo siempre son los mismos según el valor de un atributo y no del otro.

Ejemplo:

| Nº Curso | Profesor | Material |
|----------|----------|----------|
| 17       | Eva      | 1        |
| 17       | Eva      | 2        |
| 17       | Julia    | 1        |
| 17       | Julia    | 2        |
| 25       | Eva      | 1        |
| 25       | Eva      | 2        |
| 25       | Eva      | 3        |

La relación **curso**, **profesores** y **materiales del curso**.

La relación está en FNBC ya que no hay dependencias transitivas y todos los atributos son clave sin dependencia funcional hacia ellos.

Sin embargo hay **redundancia**. Los materiales se van a repetir para cualquier profesor dando cualquier curso, ya que los profesores van a utilizar todos los materiales del curso (de no ser así no habría ninguna redundancia). Los materiales del curso dependen de forma **multivaluada** del curso y no del profesor en una dependencia **multivaluada** (no hay dependencia funcional ya que los posibles valores son varios).

Para el par N° de curso y Profesor podemos saber los materiales; pero lo sabemos por el curso y no por el profesor.

- **Cuarta forma normal:**

Ocurre esta forma normal cuando una relación está en forma normal de Boyce Codd y toda dependencia multivaluada no trivial es una dependencia funcional. Son triviales aquellas dependencias multivaluadas en las que el conjunto formado por el determinante y el implicado no forman la clave primaria de la relación y además el implicado no forma parte del determinante: es decir si  $X \twoheadrightarrow Y$  y además  $Y \not\subset X$  y  $X, Y$  no es la clave de la relación, tenemos una dependencia multivaluada no trivial (como ocurre en el ejemplo anterior). Para la relación anterior la solución serían dos relaciones:

| Nº Curso | Material |
|----------|----------|
| 17       | 1        |
| 17       | 2        |
| 25       | 1        |
| 25       | 2        |
| 25       | 3        |

| Nº Curso | Profesor |
|----------|----------|
| 17       | Eva      |
| 17       | Julia    |
| 25       | Eva      |

Un teorema de Fagin indica cuando hay tres pares de conjuntos de atributos  $X$ ,  $Y$  y  $Z$  si ocurre  $X \twoheadrightarrow Y$  y  $X \twoheadrightarrow Z$  ( $Y$  y  $Z$  tienen dependencia multivaluada sobre  $X$ ),

entonces las relaciones X, Y y X, Z reproducen sin perder información lo que poseía la relación original. Este teorema marca la forma de dividir las relaciones hacia una 4FN.

## 8. Quinta forma normal (5FN)

- **Dependencias de JOIN o de reunión**

Una proyección de una **relación** es la **relación** resultante de tomar un subconjunto de los atributos de una **relación** (se trata de la operación proyección,  $\Pi$ , del álgebra relacional).

Es decir una **relación** formada por unas cuantas columnas de la relación original.

La operación **JOIN** procedente también del álgebra relacional (que luego detallaremos), consiste en formar una relación con la unión de dos relaciones.

La relación resultante estará formada por la combinación de todas las columnas y filas de ambas, excepto las columnas y filas repetidas.

Se dice que se tiene una relación con dependencia de **unión** (o de tipo JOIN) si se puede obtener esa relación como resultado de combinar mediante la operación **JOIN** varias proyecciones de la misma.

- **Quinta forma normal o forma normal de proyección-unión**

Ocurre cuando una relación está en 4FN y cada dependencia de unión (JOIN) en ella es implicada por las claves candidatas.

Es la **más compleja y polémica de todas**. Polémica pues no está claro en muchas ocasiones está muy claro que el paso a 5FN mejore la base de datos. Fue definida también por Fagin. Es raro encontrarse este tipo de problemas cuando la normalización llega a 4FN. Se deben a restricciones semánticas especiales aplicadas sobre la relación.

| Proveedor | Material | Proyecto |
|-----------|----------|----------|
| 1         | 1        | 2        |
| 1         | 2        | 1        |
| 2         | 1        | 1        |
| 1         | 1        | 1        |

Indican códigos de material suministrado por un proveedor y utilizado en un determinado proyecto. Así vista la relación, no permite ninguna **proyección** en la que no perdamos datos.

Pero si ocurre una restricción especial como por ejemplo:

***Cuando un proveedor nos ha suministrado alguna vez un determinado material, si ese material aparece en otro proyecto, haremos que el proveedor anterior nos suministre también ese material para el proyecto.***

Eso ocurre en los datos como el proveedor **número 1** nos suministró el material **número 1** para el **proyecto 2** y en el **proyecto 1** utilizamos el **material 1**, aparecerá la tupla **proveedor 1, material 1 y proyecto 1**. Si un nuevo proyecto necesitara el material 1, entonces habrá que pedirlo a los proveedores 1 y 2 (ya que en otros proyectos les hemos utilizado)

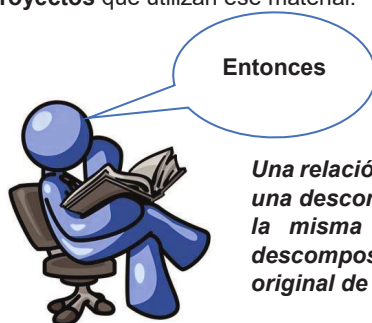
La dependencia de **reunión** que produce esta restricción es muy difícil de ver ya que es lejana. Para esa restricción esta proyección de relaciones sería válida:

| <u>Proveedor</u> | <u>Material</u> |
|------------------|-----------------|
| 1                | 1               |
| 1                | 2               |
| 2                | 1               |

| <u>Material</u> | <u>Proyecto</u> |
|-----------------|-----------------|
| 1               | 2               |
| 2               | 1               |
| 1               | 1               |

| <u>Proveedor</u> | <u>Proyecto</u> |
|------------------|-----------------|
| 1                | 2               |
| 1                | 1               |
| 2                | 1               |

Esa descomposición no pierde valores en este caso, sabiendo que si el **proveedor** nos suministra un material podremos relacionarle con todos los **proyectos** que utilizan ese material.



*Una relación no está en quinta forma normal si hay una descomposición de esa relación que muestre la misma información que la original y esa descomposición no tenga como clave la clave original de la relación.*

## 9. Forma normal de dominio clave (FNDC)

Se la conoce más con sus siglas en inglés DKNF. Se trata de una forma normal enunciada también por Fagin en 1981 al darse cuenta de los problemas de redundancia que ocurrían con algunos dominios.

En este caso no se basó en dependencias entre los datos, sino que se basó en restricciones de dominio y restricciones de clave.

- **Restricciones de dominio.** Se trata de la restricción que hace que un determinado atributo obtenga sólo ciertos valores, los que estén de acuerdo a la definición de un determinado dominio.
- **Restricción de clave.** Es la restricción que permite que un atributo o un conjunto de atributos forme una clave candidata.

Fagin dice que una relación está en FNDC si toda restricción sobre la relación es consecuencia lógica de aplicar las restricciones de dominio y clave sobre la misma. Fagin demostró que si esto ocurría la relación incluso estaba en 5FN.

| Alumno  | Nivel logrado                         | Nota |
|---------|---------------------------------------|------|
| Antonio | Muy alto y eficiente                  | 9    |
| Marisa  | Muy alto aunque sin trabajo constante | 9    |
| Mario   | Alto con trabajo habitual             | 7    |
| Luisa   | Medio aunque con trabajo              | 5    |

Observando los datos de la relación se observa que cuando la nota es superior a 9, en el nivel aparece la palabra alto, cuando es un 7 o un 8 medio, y un 5 o 6 serían medio. Es decir tenemos restricciones que **no son ni de dominio ni de clave en esa relación**. Lo lógico sería: Alumno

| Alumno  | Nivel de trabajo  | Nota |
|---------|-------------------|------|
| Antonio | Eficiente         | 9    |
| Marisa  | Trabajo constante | 9    |
| Mario   | Trabajo habitual  | 7    |
| Luisa   | Trabajo medio     | 5    |

| Nivel académico | Nota mínima | Nota máxima |
|-----------------|-------------|-------------|
| Muy alto        | 9           | 10          |
| Alto            | 7           | 8,99        |
| Medio           | 5           | 6,99        |
| Bajo            | 0           | 4,99        |

No se pierde información al diseñar las tablas de esta forma y de hecho es más eficiente para la base de datos.

## CUESTIONARIO

**Teórico:**

1. Defina con sus propias palabras que es normalización.
2. ¿Porque es importante la normalización?
3. Distinga las características principales de la primera forma normal.
4. Realice un cuadro comparativo entre la cuarta y quinta forma normal.
5. Realice un mapa mental sobre la normalización de datos.

**Investigación:**

6. ¿Quién fue Edgar Frank **Codd**; realice un mapa mental de su biografía?
7. ¿Qué es Oracle?

**Practica:**

8. Aplicar las reglas de normalización al esquema de relaciones obtenido de su proyecto bodega.

# ACTIVIDAD VI

## REPORTES DE INFORMACIÓN DE UN NEGOCIO

**OBJETIVO:** Realizado el análisis y diseño de la base de datos esta debe poder manipularse de tal manera que sirva en los procesos de negocio de la empresa para ello aplicaremos los conceptos de algebra relacional.

## ALGEBRA RELACIONAL

Es un lenguaje de consulta procedimental. Consta de un conjunto de operaciones que toman como entrada una o dos relaciones y producen como resultado una nueva relación. Las operaciones fundamentales del álgebra relacional son selección, proyección, unión, diferencia de conjuntos, producto cartesiano y renombramiento. Además de las operaciones fundamentales hay otras operaciones, por ejemplo, intersección de conjuntos, reunión natural, división y asignación. Estas operaciones se definirán en términos de las operaciones fundamentales.

### 1. Operaciones fundamentales

Las operaciones selección, proyección y renombramiento se denominan operaciones **unarias** porque operan sobre una sola relación. Las otras tres operaciones operan sobre pares de relaciones y se denominan, por lo tanto, operaciones **binarias**.

#### 1.1. La operación selección

Este operador se aplica a una relación R produciendo una nueva relación con un subconjunto de tuplas de R. La operación selección tuplas que satisfacen un predicado dado (condición).

Se utiliza la letra griega sigma minúscula ( $\sigma$ ) para denotar la selección.

El predicado (condición) aparece como subíndice de  $\sigma$ . La relación del argumento se da entre paréntesis a continuación de  $\sigma$ .

Por tanto, para seleccionar las tuplas de la relación préstamo en que la sucursal es «Navacerrada» hay que escribir

$\sigma_{\text{Nombre-sucursal}=\text{Navacerrada}}$  (préstamo)

#### Solución:

Si la relación préstamo es como:

| número-préstamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-11            | Collado Mediano | 900     |
| P-14            | Centro          | 1.500   |
| P-15            | Navacerrada     | 1.500   |
| P-16            | Navacerrada     | 1.300   |
| P-17            | Centro          | 1.000   |
| P-23            | Moralzarzal     | 2.000   |
| P-93            | Becerril        | 500     |

La relación que resulta de la consulta anterior es:

| número-préstamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-15            | Navacerrada     | 1.500   |
| P-16            | Navacerrada     | 1.300   |

Se permiten las comparaciones que utilizan =,  $\neq$ , <, ≤, > o ≥ en el predicado de selección.

- =: símbolo de igual qué.
- $\neq$ : significa no igual a, en algunos libros este símbolo esta representado por  $\neq$ .
- ≥: mayor que o igual a.

- >: mayor que.
- <: menor que.
- ≤: menor que o igual a.

**Ejemplo 1:**

Se pueden buscar todas las tuplas en las que el importe prestado sea mayor que 1.200 € escribiendo:

$\sigma_{\text{importe} > 1200}$  (préstamo)

| número-préstamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-14            | Centro          | 1.500   |
| P-15            | Navacerrada     | 1.500   |
| P-16            | Navacerrada     | 1.300   |
| P-23            | Moralzarzal     | 2.000   |

Además, se pueden combinar varios predicados en uno mayor utilizando las conectivas  $\wedge$  y  $\vee$ .

**Ejemplo 2:**

Por tanto, para encontrar las tuplas correspondientes a préstamos de más de 1.200 € concedidos por la sucursal de Navacerrada, se escribe:

$\sigma_{\text{Nombre-sucursal} = \text{"Navacerrada"} \wedge \text{importe} > 1200}$  (préstamo)

| número-préstamo | nombre-sucursal | importe |
|-----------------|-----------------|---------|
| P-15            | Navacerrada     | 1.500   |
| P-16            | Navacerrada     | 1.300   |

**Ejemplo 3:**

Se tiene la relación Ingenieros:

**Tabla Ingenieros**

| id  | nombre | edad | añosTrabajados |
|-----|--------|------|----------------|
| 123 | Leon   | 39   | 15             |
| 234 | Tomas  | 34   | 10             |
| 345 | Jose   | 45   | 21             |
| 143 | Josefa | 25   | 1              |

Seleccionar las tuplas de la tabla Ingenieros que cumplan con tener una edad mayor a 30 años:  $\sigma_{\text{edad} > 30}$  Ingenieros

| ID  | Nombre | Edad | añosTrabajados |
|-----|--------|------|----------------|
| 123 | Leon   | 39 ✓ | 15             |
| 234 | Tomas  | 34 ✓ | 10             |
| 345 | Jose   | 45 ✓ | 21             |
| 143 | Josefa | 25 ✗ | 1              |

En la imagen se ve que selecciona solo las filas que cumplen con la condición que se pedía (tener una edad mayor a 30 años), la tupla de "Josefa" queda fuera de la selección por no cumplir la condición (pues  $25 < 30$ ). De esta forma la tabla queda:

**Tabla Ingenieros**

| id  | nombre | edad | añosTrabajados |
|-----|--------|------|----------------|
| 123 | Leon   | 39   | 15             |
| 234 | Tomas  | 34   | 10             |
| 345 | Jose   | 45   | 21             |

#### Ejemplo 4:

Seleccionar de la tabla ingenieros las personas que tienen más de 30 años y que lleven menos de 16 años trabajando:

$\sigma_{\text{edad} > 30 \wedge \text{añosTrabajados} < 16}$  (Ingenieros)

| ID  | Nombre | Edad | añosTrabajados |
|-----|--------|------|----------------|
| 123 | Leon   | 39 ✓ | 15             |
| 234 | Tomas  | 34 ✓ | 10             |
| 345 | Jose   | 45 ✓ | 21             |
| 143 | Josefa | 25 ✗ | 1              |

Al tener el operador lógico AND ( $\wedge$ ) se pide que cumplan dos condiciones simultáneamente. Primero que la edad sea mayor de 30 años, al igual que en el ejemplo anterior, la tupla de "Josefa" queda fuera de la selección. Luego de las tuplas que quedan se evalúa la segunda condición. En la imagen se aprecia, que solo se seleccionan las filas que no tengan X en alguna de las condiciones.

Así finalmente quedaría la tabla:

**Tabla Ingenieros**

| id  | nombre | edad | añosTrabajados |
|-----|--------|------|----------------|
| 123 | Leon   | 39   | 15             |
| 234 | Tomas  | 34   | 10             |

### 1.2. La operación proyección

Se utiliza para producir una nueva relación desde **R**. Esta nueva relación contiene sólo algunos de los atributos de **R**, es decir, realiza la selección de algunas de las columnas de una tabla **R**.

La proyección se denota por la letra griega mayúscula pi ( $\Pi$ ).

**Supóngase** que se desea hacer una lista de todos los números de préstamo y del importe de los mismos, pero sin que aparezcan los nombres de las sucursales. La operación proyección permite producir esta relación.

$\Pi$ número-préstamo, importe (**préstamo**)

La relación que resulta de esta consulta se muestra:

| número-préstamo | importe |
|-----------------|---------|
| P-11            | 900     |
| P-14            | 1.500   |
| P-15            | 1.500   |
| P-16            | 1.300   |
| P-17            | 1.000   |
| P-23            | 2.000   |
| P-93            | 500     |

**Ejemplo 1:**

Dada la tabla:

**Tabla Ingenieros**

| id  | nombre | edad | añosTrabajados |
|-----|--------|------|----------------|
| 123 | Leon   | 39   | 15             |
| 234 | Tomas  | 34   | 10             |
| 345 | Jose   | 45   | 21             |
| 143 | Josefa | 25   | 1              |

Mostrar las columnas de id y nombre de la tabla de ingenieros.

$\pi$  (id,nombre) **Ingenieros**

**Tabla Ingenieros**

| id  | nombre |
|-----|--------|
| 123 | Leon   |
| 234 | Tomas  |
| 345 | Jose   |
| 143 | Josefa |

**Ejemplo 2:**

Mostrar el id y nombre de los Ingenieros que tienen más de 30 años.

$\pi$  (id,nombre) ( $\sigma_{\text{edad} > 30}$  **Ingenieros**)

| ID  | Nombre | Edad | añosTrabajados |
|-----|--------|------|----------------|
| 123 | Leon   | 39 ✓ | 15             |
| 234 | Tomas  | 34 ✓ | 10             |
| 345 | Jose   | 45 ✓ | 21             |
| 143 | Josefa | 25 ✗ | 1              |

Se aprecia que las tuplas que no cumplan con la condición de selección quedan fuera del resultado, luego se realiza un Project sobre las filas del resultado, separando solo las columnas que contienen los atributos id y nombre. Finalmente la tabla queda de la siguiente manera:

## Tabla Ingenieros

| id  | nombre |
|-----|--------|
| 123 | Leon   |
| 234 | Tomas  |
| 345 | Jose   |

### 1.3. La operación unión

Considérese una consulta para averiguar el nombre de todos los clientes del banco que tienen una cuenta, un préstamo o ambas cosas. Obsérvese que la relación **cliente** no contiene esa información, dado que los clientes no necesitan tener ni cuenta ni préstamo en el banco.

Para contestar a esta consulta hace falta la información de la relación **impositor** y la de la relación **prestatario**

#### Impositor

| nombre cliente | número cuenta |
|----------------|---------------|
| Abril          | C-102         |
| Gómez          | C-101         |
| González       | C-201         |
| González       | C-217         |
| López          | C-222         |
| Rupérez        | C-215         |
| Santos         | C-305         |

#### Prestario

| nombre cliente | número préstamo |
|----------------|-----------------|
| Fernández      | P-16            |
| Gómez          | P-93            |
| Gómez          | P-15            |
| López          | P-14            |
| Pérez          | P-17            |
| Santos         | P-11            |
| Sotoca         | P-23            |
| Valdivieso     | P-17            |

Se conoce la manera de averiguar los nombres de todos los clientes con préstamos en el banco:

También se conoce la manera de averiguar el nombre de los clientes con cuenta en el banco:

$$\Pi_{\text{nombre-cliente}}(\text{impositor})$$

**Para contestar** a la consulta hace falta la **unión** de estos dos conjuntos; es decir, hacen falta todos los nombres de clientes que aparecen en alguna de las dos relaciones o en ambas. Estos datos se pueden averiguar mediante la operación binaria **unión**, denotada, como en la teoría de conjuntos, por  $\cup$ . Por tanto, la expresión buscada es:

Téngase en cuenta que en el resultado hay diez tuplas, aunque hay siete prestatarios y seis impositores distintos. Esta discrepancia aparente se debe a que Gómez, Santos y López son a la vez prestatarios e impositores. Dado que las relaciones son conjuntos, se eliminan los valores duplicados. Obsérvese que en este ejemplo se toma la unión de dos conjuntos, ambos consistentes en valores de nombre-cliente. En general, se debe asegurar que las uniones se realicen entre relaciones compatibles.

| nombre-cliente |
|----------------|
| Abril          |
| Fernández      |
| Gómez          |
| González       |
| López          |
| Pérez          |
| Rupérez        |
| Santos         |
| Sotoca         |
| Valdivieso     |

Por tanto, para que una operación unión **RUS** sea válida hay que exigir que se cumplan dos condiciones:

1. Las relaciones *r* y *s* deben ser de la misma paridad. Es decir, deben tener el mismo número de atributos.
2. Los dominios de los atributos *i*-ésimos de **R** y de **S** deben ser iguales para todo *i*.

#### 1.4. La operación diferencia de conjuntos

La operación diferencia de conjuntos, denotada por  $-$ , permite buscar las tuplas que estén en una relación pero no en la otra. La expresión  $R - S$  da como resultado una relación que contiene las tuplas que están en *R* pero no en *S*.

Se pueden buscar todos los **clientes del banco** que tienen abierta una cuenta pero no tienen concedido ningún préstamo escribiendo:

| nombre-cliente |
|----------------|
| Abril          |
| González       |
| Rupérez        |

Como en el caso de la operación **unión**, hay que asegurarse de que las **diferencias de conjuntos** se realicen entre relaciones compatibles. Por tanto, para que una operación diferencia de conjuntos  $R - S$  sea válida hay que exigir que las relaciones **R Y S** sean de la misma paridad y que los dominios de los atributos *i*-ésimos de *r* y *s* sean iguales.

### 1.5. La operación producto cartesiano

La operación producto cartesiano, denotada por un aspa ( $\times$ ), permite combinar información de cualesquiera dos relaciones.

El producto cartesiano de las relaciones **r1** y **r2** como **r1  $\times$  r2**.

Recuérdese que las relaciones se definen como subconjuntos del producto cartesiano de un conjunto de dominios. A partir de esta definición ya se debe tener una intuición sobre la definición de la operación producto cartesiano.

**Sin embargo**, dado que el mismo nombre de atributo puede aparecer tanto en **r1** como en **r2**, hay que crear un esquema de denominaciones para distinguir entre ambos atributos. En este caso se logra adjuntando al atributo el nombre de la relación de la que proviene originalmente.

**Por ejemplo**, el esquema de relación de **r = prestatario  $\times$  préstamo** es:

**(prestatario.nombre-cliente, prestatario.númeropréstamo,  
préstamo.nombre-sucursal, préstamo.número-préstamo,  
préstamo.importe)**

Con este esquema se puede distinguir entre prestatario.número-préstamo y préstamo.número-préstamo.

**Para los atributos** que sólo aparecen en uno de los dos esquemas se suele omitir el prefijo con el nombre de la **relación**. Esta simplificación no genera ambigüedad alguna.

**Por tanto**, se puede escribir el esquema de relación de **r** como:

(nombre-cliente, prestatario.número-préstamo, nombre-sucursal,  
préstamo.número-préstamo, importe)

El acuerdo de denominaciones precedente **exige que las relaciones** que sean argumentos de la operación producto cartesiano tengan nombres diferentes.

Ahora que se conoce el esquema de relación de **r = prestatario  $\times$  préstamo** hay que averiguar las tuplas que aparecerán en **r**. Como se podía imaginar, se crea una tupla de **r** a partir de cada par de tuplas posible: una de la relación prestatario y otra de la relación préstamo. Por tanto, **r** es una relación de gran tamaño:

| nombre-cliente | prestatario.número-préstamo | préstamo.número-préstamo | nombre-sucursal | importe |
|----------------|-----------------------------|--------------------------|-----------------|---------|
| Santos         | P-17                        | P-11                     | Collado Mediano | 900     |
| Santos         | P-17                        | P-14                     | Centro          | 1.500   |
| Santos         | P-17                        | P-15                     | Navacerrada     | 1.500   |
| Santos         | P-17                        | P-16                     | Navacerrada     | 1.300   |
| Santos         | P-17                        | P-17                     | Centro          | 1.000   |
| Santos         | P-17                        | P-23                     | Moralzarzal     | 2.000   |
| Santos         | P-17                        | P-93                     | Becerril        | 500     |
| Gómez          | P-23                        | P-11                     | Collado Mediano | 900     |
| Gómez          | P-23                        | P-14                     | Centro          | 1.500   |
| Gómez          | P-23                        | P-15                     | Navacerrada     | 1.500   |
| Gómez          | P-23                        | P-16                     | Navacerrada     | 1.300   |
| Gómez          | P-23                        | P-17                     | Centro          | 1.000   |
| Gómez          | P-23                        | P-23                     | Moralzarzal     | 2.000   |
| Gómez          | P-23                        | P-93                     | Becerril        | 500     |
| López          | P-15                        | P-11                     | Collado Mediano | 900     |
| López          | P-15                        | P-14                     | Centro          | 1.500   |
| López          | P-15                        | P-15                     | Navacerrada     | 1.500   |
| López          | P-15                        | P-16                     | Navacerrada     | 1.300   |
| López          | P-15                        | P-17                     | Centro          | 1.000   |
| López          | P-15                        | P-23                     | Moralzarzal     | 2.000   |
| López          | P-15                        | P-93                     | Becerril        | 500     |
| ...            | ...                         | ...                      | ...             | ...     |
| ...            | ...                         | ...                      | ...             | ...     |
| Valdivieso     | P-17                        | P-11                     | Collado Mediano | 900     |
| Valdivieso     | P-17                        | P-14                     | Centro          | 1.500   |
| Valdivieso     | P-17                        | P-15                     | Navacerrada     | 1.500   |
| Valdivieso     | P-17                        | P-16                     | Navacerrada     | 1.300   |
| Valdivieso     | P-17                        | P-17                     | Centro          | 1.000   |
| Valdivieso     | P-17                        | P-23                     | Moralzarzal     | 2.000   |
| Valdivieso     | P-17                        | P-93                     | Becerril        | 500     |
| Fernández      | P-16                        | P-11                     | Collado Mediano | 900     |
| Fernández      | P-16                        | P-14                     | Centro          | 1.500   |
| Fernández      | P-16                        | P-15                     | Navacerrada     | 1.500   |
| Fernández      | P-16                        | P-16                     | Navacerrada     | 1.300   |
| Fernández      | P-16                        | P-17                     | Centro          | 1.000   |
| Fernández      | P-16                        | P-23                     | Moralzarzal     | 2.000   |
| Fernández      | P-16                        | P-93                     | Becerril        | 500     |

Supóngase que se tienen  $n_1$  tuplas en prestatario y  $n_2$  tuplas en préstamo. Por tanto, hay  $n_1 * n_2$  maneras de escoger un par de tuplas, una tupla de cada relación; por lo que hay  $n_1 * n_2$  tuplas en  $r$ . En concreto, obsérvese que para algunas tuplas  $t$  de  $r$  puede ocurrir que  $t[\text{prestatario.número-préstamo}] \neq t[\text{préstamo.número-préstamo}]$ .

En general, si se tienen las relaciones  $r_1 (R_1)$  y  $r_2 (R_2)$ ,  $r_1 \times r_2$  es una relación cuyo esquema es la concatenación de  $R_1$  y de  $R_2$ . La relación  $R$  contiene todas las tuplas  $t$  para las que hay unas tuplas  $t_1$  en  $r_1$  y  $t_2$  en  $r_2$  para las que  $t[R_1] = t_1[R_1]$  y  $t[R_2] = t_2[R_2]$ .

Supóngase que se desea averiguar los nombres de todos los clientes que tienen concedido un préstamo en la sucursal de Navacerrada. Se necesita para ello información de las relaciones préstamo y prestatario. Si se escribe:

| nombre-cliente | prestatario.número-préstamo | préstamo.número-préstamo | nombre-sucursal | importe |
|----------------|-----------------------------|--------------------------|-----------------|---------|
| Santos         | P-17                        | P-15                     | Navacerrada     | 1.500   |
| Santos         | P-17                        | P-16                     | Navacerrada     | 1.300   |
| Gómez          | P-23                        | P-15                     | Navacerrada     | 1.500   |
| Gómez          | P-23                        | P-16                     | Navacerrada     | 1.300   |
| López          | P-15                        | P-15                     | Navacerrada     | 1.500   |
| López          | P-15                        | P-16                     | Navacerrada     | 1.300   |
| Sotoca         | P-14                        | P-15                     | Navacerrada     | 1.500   |
| Sotoca         | P-14                        | P-16                     | Navacerrada     | 1.300   |
| Pérez          | P-93                        | P-15                     | Navacerrada     | 1.500   |
| Pérez          | P-93                        | P-16                     | Navacerrada     | 1.300   |
| Gómez          | P-11                        | P-15                     | Navacerrada     | 1.500   |
| Gómez          | P-11                        | P-16                     | Navacerrada     | 1.300   |
| Valdivieso     | P-17                        | P-15                     | Navacerrada     | 1.500   |
| Valdivieso     | P-17                        | P-16                     | Navacerrada     | 1.300   |
| Fernández      | P-16                        | P-15                     | Navacerrada     | 1.500   |
| Fernández      | P-16                        | P-16                     | Navacerrada     | 1.300   |

Se tiene una relación que sólo atañe a la sucursal de Navacerrada. Sin embargo, la columna nombrecliente puede contener clientes que no tengan concedido ningún préstamo en la sucursal de Navacerrada. (Si no se ve el motivo por el que esto es cierto, recuérdese que el producto cartesiano toma

todos los emparejamientos posibles de una tupla de prestatario con una tupla de préstamo.)

Dado que la operación producto cartesiano asocia todas las tuplas de **préstamo** con todas las tuplas de **prestatario**, se sabe que, si un cliente tiene concedido un préstamo en la sucursal de **Navacerrada**, hay alguna tupla de **prestatario × préstamo** que contiene su nombre y que prestatario.número-préstamo= préstamo.número-préstamo.

Por tanto, si escribimos:

Sólo se obtienen las tuplas de prestatario × préstamo que corresponden a los clientes que tienen concedido un préstamo en la sucursal de Navacerrada. Finalmente, dado que sólo se desea obtener nombrecliente, se realiza una proyección:

El resultado es:

| <i>nombre-cliente</i> |
|-----------------------|
| Fernandez             |
| López                 |

### 1.6. Operación renombramiento

A diferencia de las relaciones de la base de datos, los resultados de las expresiones de álgebra relacional no tienen un nombre que se pueda utilizar para referirse a ellas. Resulta útil poder ponerles nombre; el operador renombramiento, denotado por la letra griega **rho** minúscula (  $\rho$  ), permite realizar esta tarea.

Dada una expresión E del álgebra relacional, la expresión

$$\rho_x(E)$$

devuelve el resultado de la expresión E con el nombre x.

**Las relaciones r por sí mismas** se consideran expresiones (triviales) del álgebra relacional. Por tanto, también se puede aplicar la operación renombramiento a una relación r para obtener la misma relación con un nombre nuevo.

**Para ilustrar el uso del renombramiento** de las relaciones, considérese la consulta «**Buscar el máximo saldo de cuenta del banco**».

La estrategia empleada para obtener el resultado es

- 1) calcular una relación intermedia consistente en los saldos que no son el máximo y
- 2) realizar la diferencia entre la relación saldo (cuenta) y la relación intermedia recién calculada.

**Paso 1:** Para calcular la relación intermedia hay que comparar los valores de los saldos de todas las cuentas. Esta comparación se puede hacer calculando el producto cartesiano  $\text{cuenta} \times \text{cuenta}$  y formando una selección para comparar el valor de cualesquiera dos saldos que aparezcan en una tupla.

En primer lugar hay que crear un mecanismo para distinguir entre los dos atributos saldo. Se utilizará la operación renombramiento para cambiar el nombre de una referencia a la relación cuenta; así, se puede hacer referencia dos veces a la relación sin ambigüedad alguna.

La relación temporal que se compone de los saldos que no son el máximo puede escribirse ahora como

Esta expresión proporciona los saldos de la relación cuenta para los que aparece un saldo mayor en alguna parte de la relación cuenta (cuyo nombre se ha cambiado a d). El resultado contiene todos los saldos salvo el máximo.

| saldo |
|-------|
| 500   |
| 400   |
| 700   |
| 750   |
| 350   |

**Paso 2:** La consulta para averiguar el máximo saldo de cuenta del banco puede escribirse de la manera siguiente:

**Resultado:**

| saldo |
|-------|
| 900   |

**Ejemplo:**

«Averiguar los nombres de todos los clientes que viven en la misma calle y en la misma ciudad que Gómez». Se puede obtener la calle y la ciudad en la que vive Gómez escribiendo:

Sin embargo, para hallar a otros clientes que vivan en esa calle y en esa ciudad hay que hacer referencia por segunda vez a la relación cliente. En la consulta siguiente se utiliza la operación renombramiento sobre la expresión anterior para darle al resultado el nombre dirección-Gómez y para cambiar el nombre de los atributos a calle y ciudad en lugar de calle-cliente y ciudad-cliente:

**Resultado:**

| <i>nombre-cliente</i> |
|-----------------------|
| Gómez<br>Pérez        |

## CUESTIONARIO

**Teórico:**

1. ¿Por qué es importante el álgebra relacional?
2. ¿Cuál es la diferencia entre la operación selección y proyección?
3. Describa el funcionamiento de la operación producto cartesiano

**Investigación:**

4. ¿Existen otras operaciones del álgebra relacional; menciónalas e indica una característica importante?
5. ¿Qué es SQL?

**Practica:**

6. Se tienen las siguientes relaciones:

**ADMINISTRADOR**

| código | nombre        | edad |
|--------|---------------|------|
| 1      | Jorge Campos  | 33   |
| 2      | Enrique Muñoz | 25   |
| 3      | Esteban Paz   | 21   |

**PRODUCTOR**

| código | nombre        | edad |
|--------|---------------|------|
| 2      | Enrique Muñoz | 25   |
| 8      | Jorge Arias   | 30   |
| 10     | Juan Martínez | 19   |

- Seleccionar los administradores cuya edad sea mayor 24
- Seleccionar al productor cuyo código es 2
- Mostrar el nombre y edad de los administradores.

7. En la relación:

**JUGADOR**

| cod | nom  | país  |
|-----|------|-------|
| 3   | Xiu  | China |
| 8   | Juan | Perú  |
| 2   | Juan | Perú  |

- Mostrar cada jugador con su respectivo país.
  - Mostrar los jugadores de Perú
8. Crea el esquema de la base de datos “proyecto bodega”, y crea ejemplos de selección, proyección y diferencia.

# ACTIVIDAD VII

## **DIAGRAMACIÓN DEL MODELO DE BASE DE DATOS DEL NEGOCIO USANDO COMO HERRAMIENTA TECNOLÓGICA ERWIN**

OBJETIVO: Utilizar Erwin para modelar la base de datos (lógico y físico) y optimizar el sistema de información de la empresa.

## INTRODUCCIÓN A Erwin 7.1

Es una herramienta para el diseño de base de datos, que Brinda productividad en su diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada, además ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos. Genera automáticamente las tablas y miles de líneas de stored procedure y triggers para los principales tipos de base de datos.

ERwin hace fácil el diseño de una base de datos. Los diseñadores de bases de datos sólo apuntan y pulsan un botón para crear un gráfico del modelo E-R (Entidad \_ relación) de todos sus requerimientos de datos y capturar las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos, relaciones, y llaves importantes.

La migración automática garantiza la integridad referencial de la base de datos. ERwin establece una conexión entre una base de datos diseñada y una base de datos, permitiendo transferencia entre ambas y la aplicación de ingeniería reversa. Usando esta conexión, ERwin genera automáticamente tablas, vistas, índices, reglas de integridad referencial (llaves primarias, llaves foráneas), valores por defecto y restricciones de campos y dominios.

ERwin soporta principalmente bases de datos relacionales SQL y bases de datos que incluyen Oracle, Microsoft SQL Server, Sybase. El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.

### 1. Conceptos básicos.

- **Base de Datos** Conjunto de datos organizados de tal manera, que pueda extraerse información y que pueda ser compartida. Entre los usuarios debe existir un consenso en lo que representan los datos (la información a obtener).
- **Modelo de Datos** Estructuras de datos y reglas de negocio que representan los requerimientos de un sistema.
- **Entidad** Una persona, lugar, evento o concepto acerca del cual el negocio necesita guardar datos.
- **Atributo** Propiedad de una entidad que almacenara datos.
- **Llave Primaria (PK)** Un atributo (Llave simple) o conjunto de atributos (Llave compuesta) que identifican únicamente una instancia (fila o registro) de una entidad. En ERwin la llave primaria está posicionada sobre la línea en una entidad.
- **Llave Alterna (AK)** Un atributo (Llave simple) o conjunto de atributos (Llave compuesta) que identifican únicamente una instancia (fila o registro) de una entidad, pero que NO ES ESCOGIDA como llave primaria. En ERwin, se muestra por el diagrama (AKx,y) donde x representa un numero entero incrementado para cada llave alterna en una entidad; y representa el orden del atributo llave.
- **Entrada Inversa (IE)** Se utilizan cuando uno o más atributos son usados frecuentemente para acceder a una o más instancias (filas o registros) de las entidades. EN ERwin son mostrados en diagramas (IEx,y) después de cada atributo de la entrada inversa. donde x representa un número entero incrementado para cada llave alterna en una entidad; y representa el orden del atributo llave.

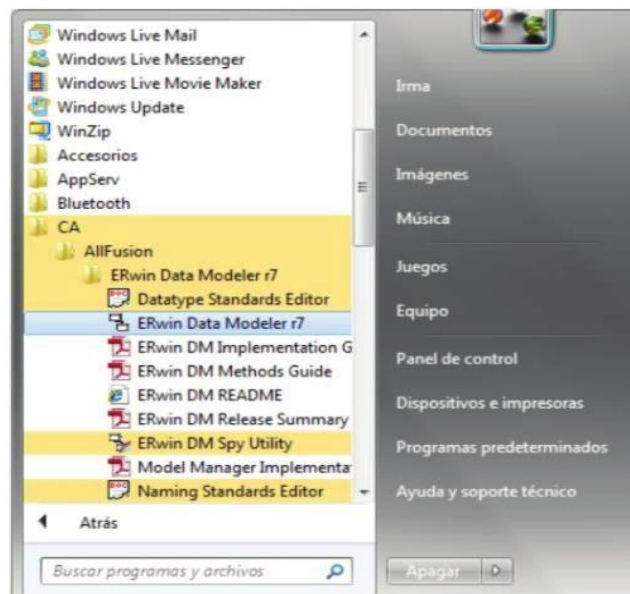
- **Relación** Un enlace lógico entre dos entidades que representa una regla de negocio o una restricción.
- **Llave Foránea (FK)** Una llave foránea es una llave primaria de una entidad padre (Fuerte) que es AGREGADA a la entidad hijo (Débil) a través de su relación.
- **Relación Identificada** La llave primaria de la entidad padre es migrada a través de la RELACION para FORMAR parte de la llave primaria de la entidad hijo.
- **Relación Obligatoria** No-Identificada La llave primaria de la entidad padre es migrada como un atributo no llave (no forma parte de la llave primaria de la entidad hijo) de la entidad hijo. La FK necesariamente tiene que tener un valor real de la PK.
- **Relación No-Obligatoria** No-Identificada La llave primaria de la entidad padre es migrada como un atributo no llave (no forma parte de la llave primaria de la entidad hijo) de la entidad hijo. La opcionalidad en el lado del padre indica que la FK en la entidad hijo puede existir sin la información de la PK de la entidad padre.
- **Relación Muchos a Muchos** La llave primaria de la entidad padre no es migrada como llave foránea. Cada frase representa la regla desde la perspectiva ·Padre a Hijo ·Hijo a Padre

## 2. Descripción del entorno

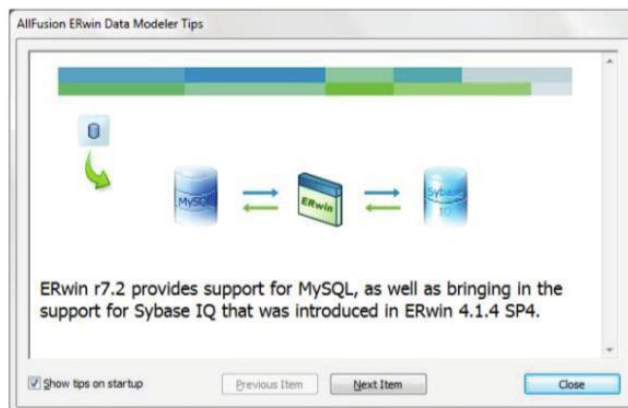
Para conocer el uso de las herramientas del Erwin 7.2 primero ingresaremos al Erwin;



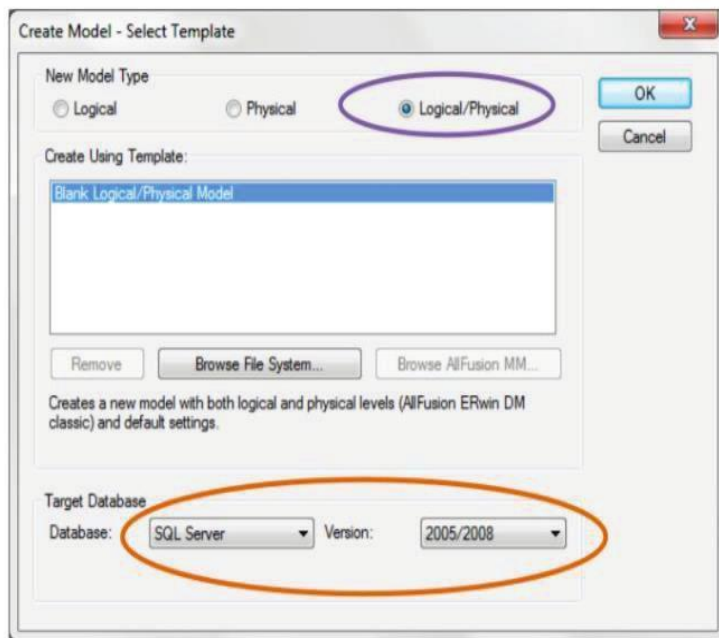
- Entrar en  buscar en todos los programas
- Escoger Erwin Data Modeler r7



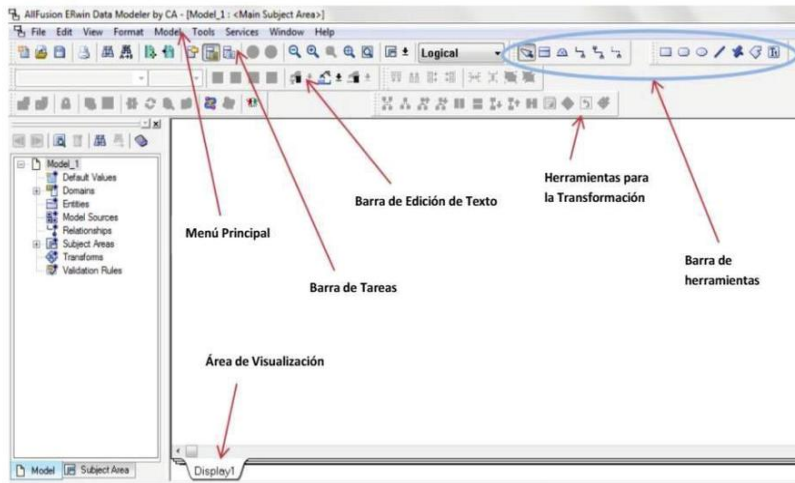
- La siguiente ventana nos indica el soporte que provee el Erwin r7.2
- Cerramos la ventana



- Se puede escoger el tipo de modelo, normalmente se trabaja con el modelo lógico y físico.
- Escogemos la base de datos con que se trabajara
- Le damos ok



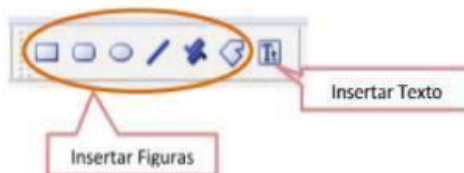
## La pantalla principal de Erwin.



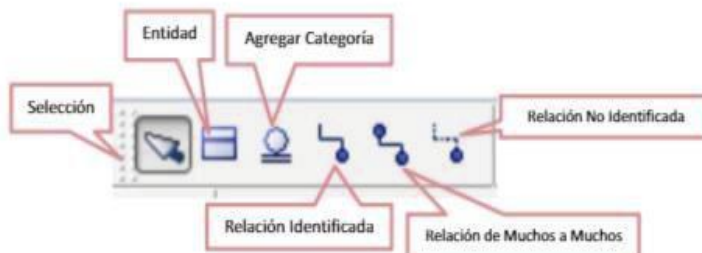
### ➤ La barra de herramientas de Erwin r7.2



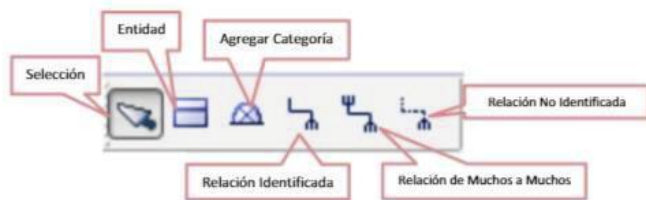
### ➤ Herramientas para Editar



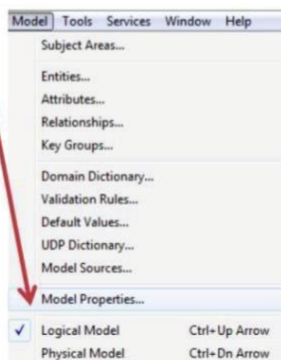
### ➤ Barra de Herramientas Lógica IDEF1X



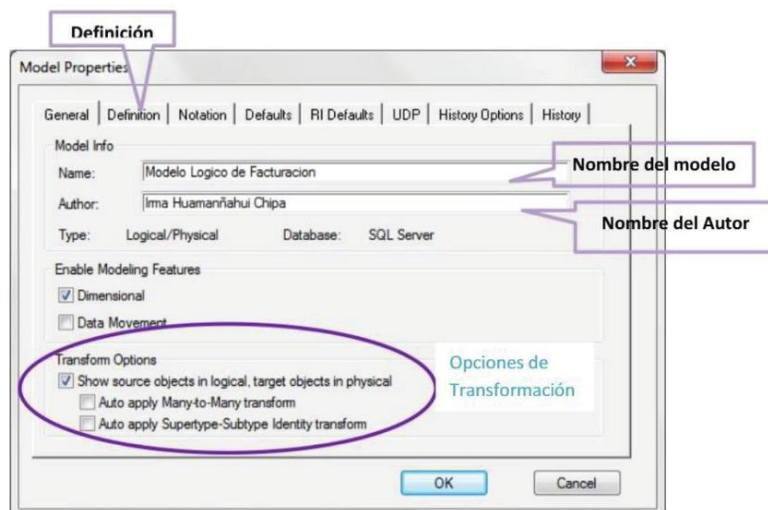
## ➤ Barra de Herramientas Lógica IE



- Entramos en el menú principal
- Seleccionamos Model
- Seleccionamos



## ➤ Propiedades del modelo



### 3. Creando Entidades y Atributos

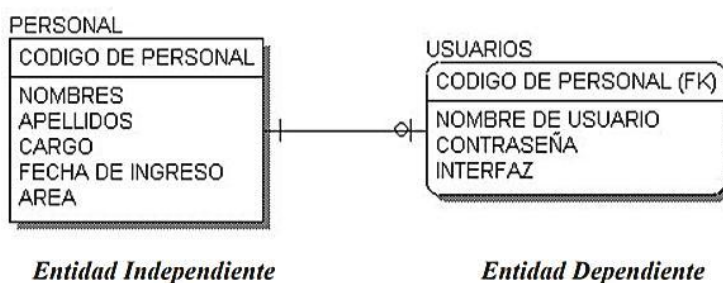
#### 3.1. Construyendo Entidades

En ERwin el modelo lógico puede contener dos tipos de entidades: **independiente y dependiente**.

Una **entidad independiente** es una entidad que puede identificarse sin determinar su relación con otra entidad. Cada entidad tiene llave propia, se representa como una caja con rincones cuadrados.

Una **entidad dependiente** es una entidad que para identificarse requiere de su relación a otra entidad o entidades. Se representa como una caja con rincones redondeados.

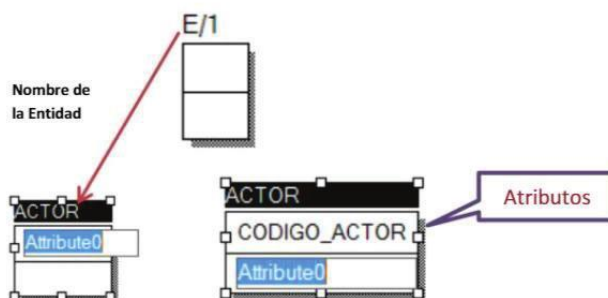
**Ejemplo:**



**Procedemos a crear entidades:**

- Nos ubicamos en la barra de herramientas

- Hacemos clic en



- Si se desea se puede editar el texto en las entidades
- Nos ubicamos en la barra de edición de texto



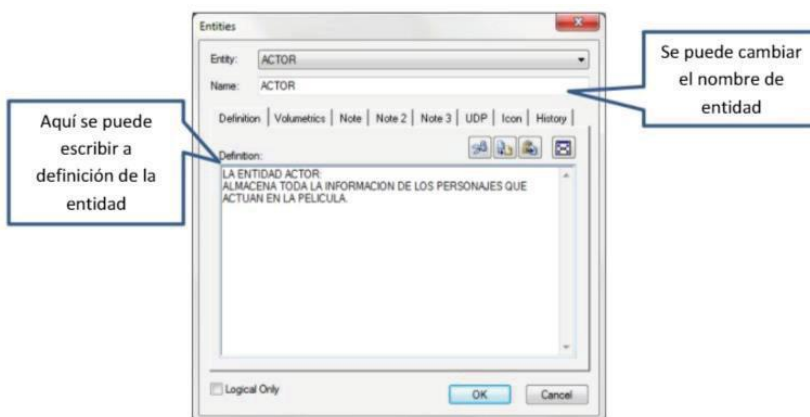
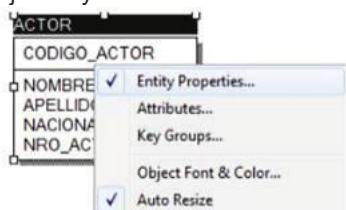
### 3.2. Editar entidades:



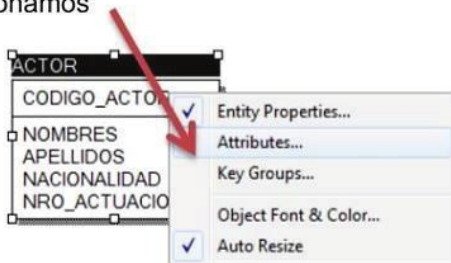
#### Editar

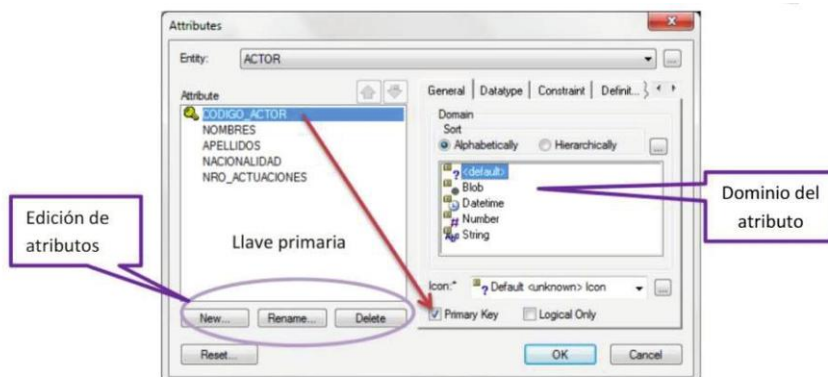
Utilice el editor de entidades para ingresar/editar, definiciones de entidades y sus notas, para explorar definiciones, cambiar el nombre a la entidad o para asignar propiedades definidas por el usuario UPD. **Para ello realizar lo siguiente:**

- Clic derecho en una Entidad para mostrar el menú contextual.
- Escoja Entity Editor



- Para definir los atributos: hacemos clic derecho en la entidad
- seleccionamos



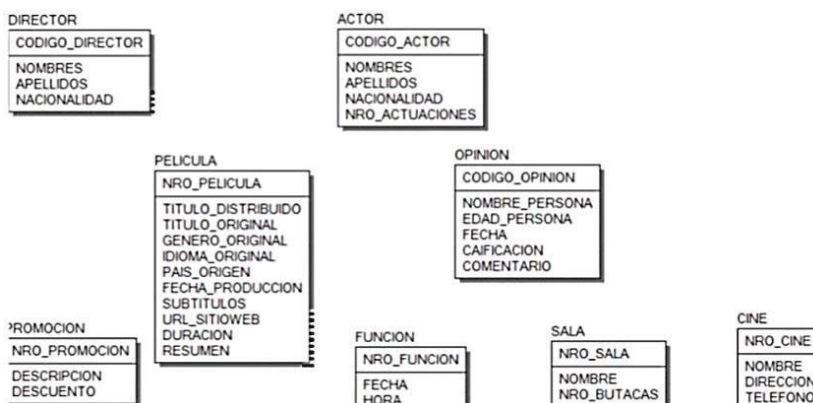


#### 4. Elaboramos: Modelo Cine

##### 4.1. Entidades y atributos

- **DIRECTOR:** Almacena toda la información de los encargados de la dirección de la película durante su grabación y producción.
- **ACTOR:** Almacena información de los personajes que actualmente actual en la película.
- **PELICULA:** Contiene información referente a la película.
- **OPINION:** Contiene los comentarios de las personas pueden ser positivas o negativos.
- **PROMOCIÓN:** Guardaremos información con respecto a las promociones que se les brinda a los usuarios; pueden ser regalos, descuentos, etc.
- **FUNCIÓN:** Almacena información referente a horarios de presentación de películas.
- **SALA:** Almacena información respecto a donde se llevara a cabo la función
- **CINE:** Almacena información del lugar donde se llevara a cabo la película.

##### Atributos:



#### 4.2. Creación de llaves.

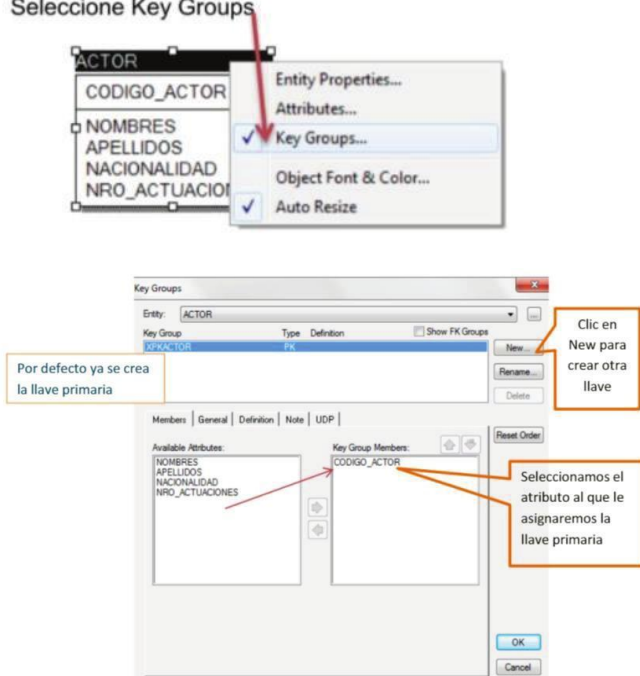
- **Clave primaria (PK).**

Por definición un Llave Primaria es aquella que identifica de manera única una fila de la entidad. Esta debe cumplir ciertos requisitos:

- No cambia con el tiempo
- No acepta valores NULL
- No acepta valores repetidos

En ERwin para crear una Llave Primaria realizar lo siguiente:

- Clic derecho sobre la entidad
- Seleccione Key Groups

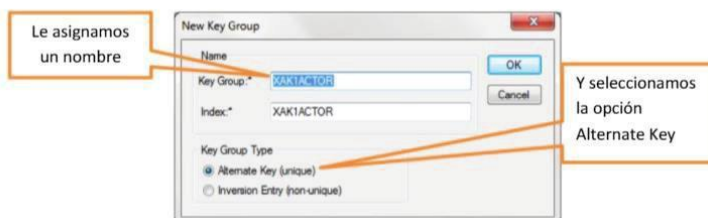


- **Llave alterna (AK)**

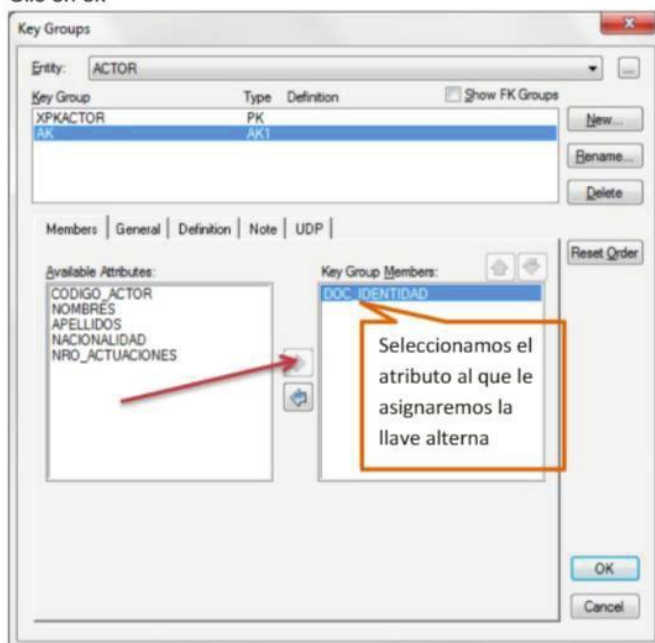
Es un atributo o grupos de atributos que no fueron elegidos como Llaves Primarias. En ERwin para crear una Llave Alterna realizar lo siguiente:

1. Clic derecho sobre la entidad.
2. Seleccione Key Group...
3. ERwin le mostrara la siguiente ventana

➤ Resulta la siguiente ventana



➤ Clic en ok

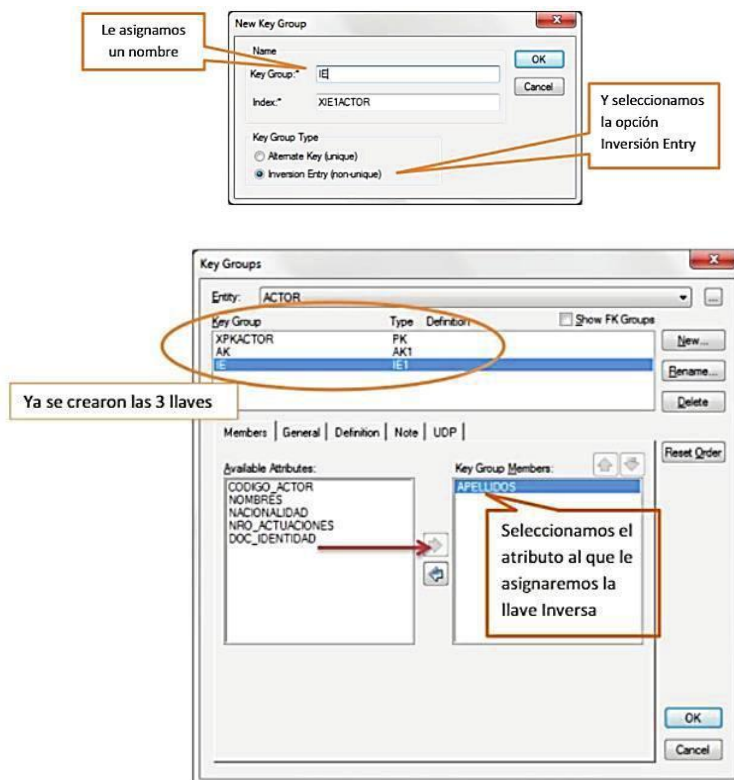


➤ Clic en ok

- **Llave inversa (IE)**

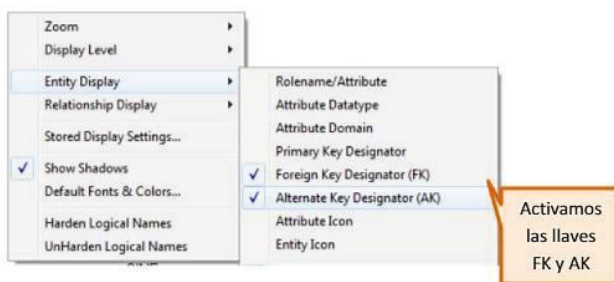
Para crear Llaves Inversa o Entrada Inversa repita los mismos pasos que siguió para crear una Llave Alterna.

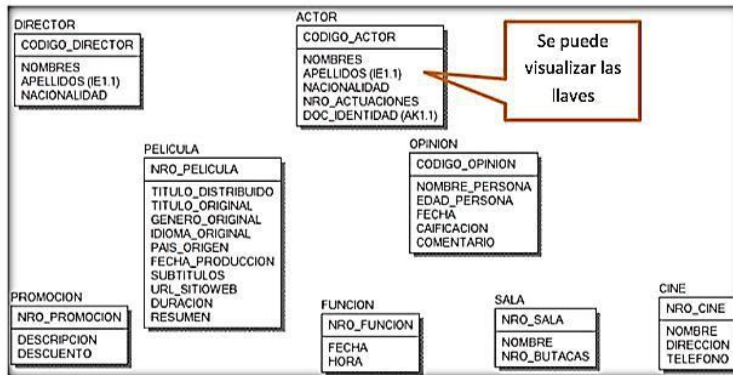
En la ventana New Key Group elija el botón que tiene como título Inversion Entry (no-unique). Tal como se muestra en la figura:



Para visualizar la creación de las llaves en el modelo, realizamos lo siguiente:

1. Clic derecho sobre el modelo
2. Selecciona **Entity Display**





#### 4.3. Relaciones

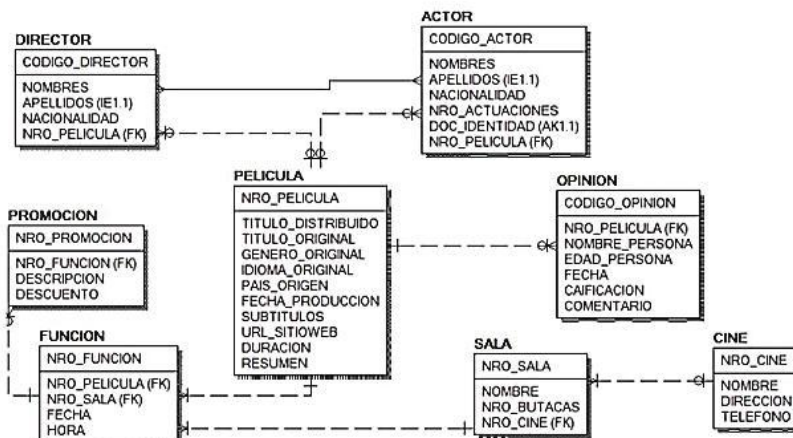
ERwin define tres tipos de relaciones:

- Relación Identificada
- Relación no Identificada
- Relación Muchos a Muchos

Para todos los casos la construcción de las relaciones es tal como se detalla a continuación:

1. Seleccione el tipo de relación de la caja de herramienta.
2. Seleccione la entidad Padre.
3. Seleccione la entidad hijo.
4. Seleccione la flecha de la caja de herramienta para desactivar la relación.

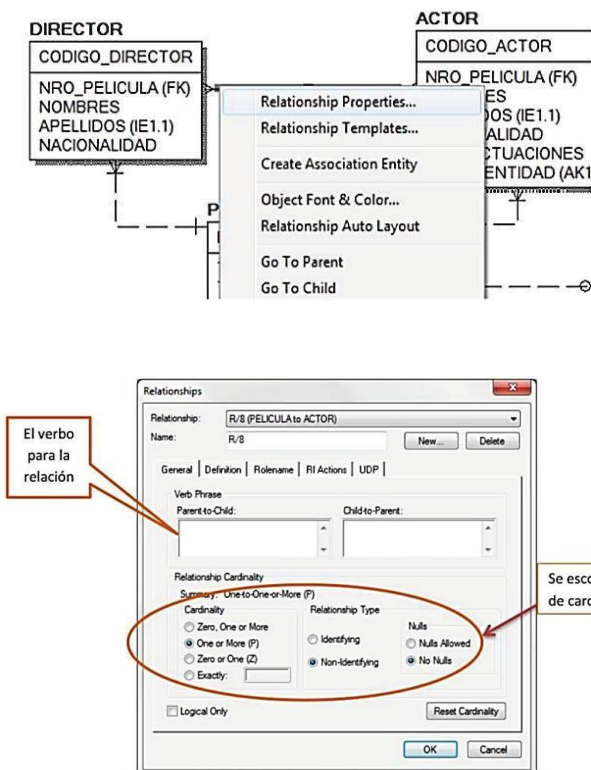
#### MODELO DE PROGRAMACIÓN DE PELICULAS EN UN CINE



- **Editando la Relaciones**

Para editar una relación realizar lo siguiente:

1. Haga doble clic sobre la relación o anticlic y seleccione propiedades de relaciones.
2. ERwin le mostrara la siguiente ventana:



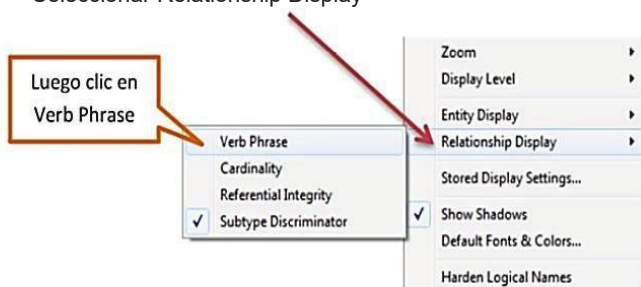
- **Creando Frase-Verbo a las Relaciones**

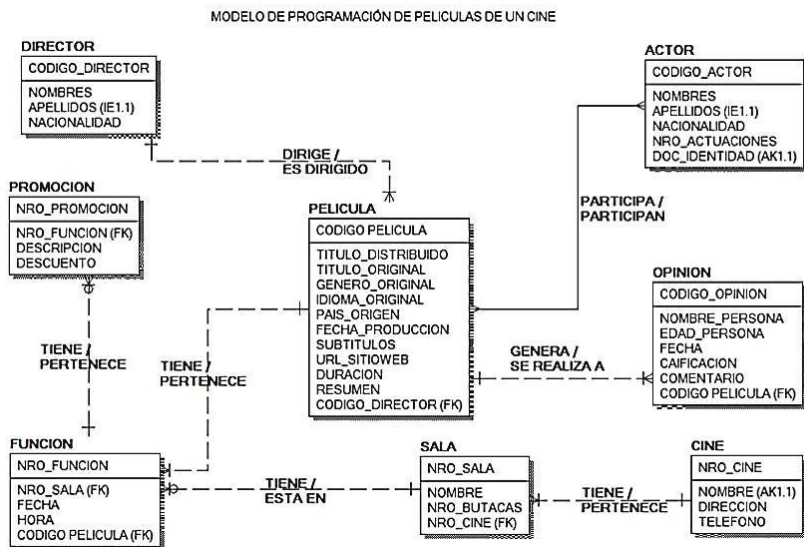
Una frase verbo permite describir la relación entre dos entidades, para ello realizar lo siguiente:

1. Doble clic sobre la relación.
2. En el TAB General ingrese a la zona que tiene como título Parent-to-Child y escriba el verbo para la relación.
3. Finalmente clic en OK.

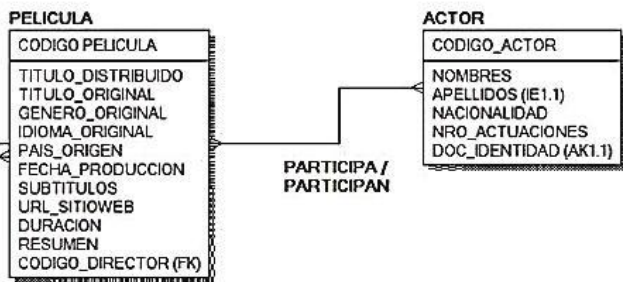
Para visualizar las relaciones debemos hacer:

1. Anti clic sobre el modelo
2. Seleccionar Relationship Display

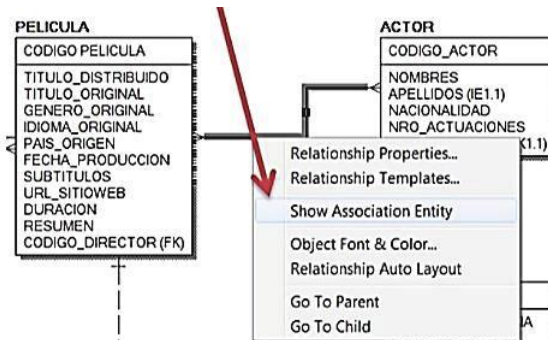




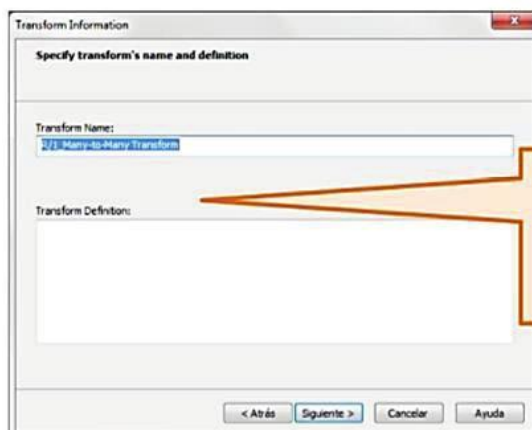
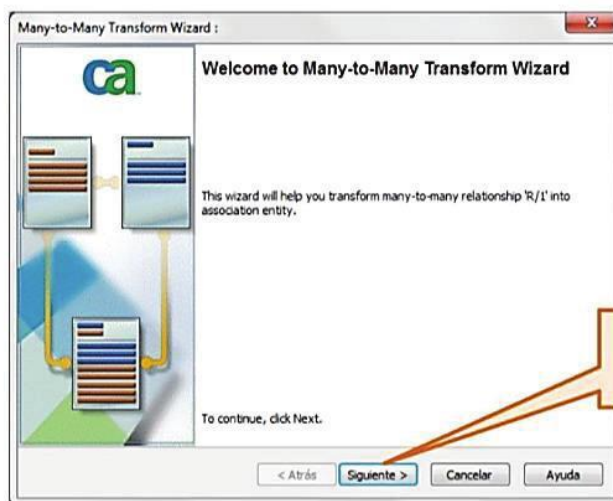
Cuando existe la relación de muchos a muchos se tiene que generar una nueva entidad.



1. Hacer anticlic en la relación de muchos a muchos
2. Seleccionamos Show Association Entity



➤ Resulta la siguiente ventana:



Association Entity Information

Specify the name and definition for association entity

Entity Name:  
DIRECTOR\_ACTOR

Entity Definition:

< Atrás   **Siguiente >**   Cancelar   Ayuda

Le  
asignamos  
un nombre

Summary

Completing the Many-to-Many Transform Wizard

You have successfully completed the steps needed to create Many-to-Many Transform.

Summary of changes:  
You will create Many-to-Many Transform R/I\_Many-to-Many Transform.  
It will transform the many to many relationship: R/I into the association entity: DIRECTOR\_ACTOR

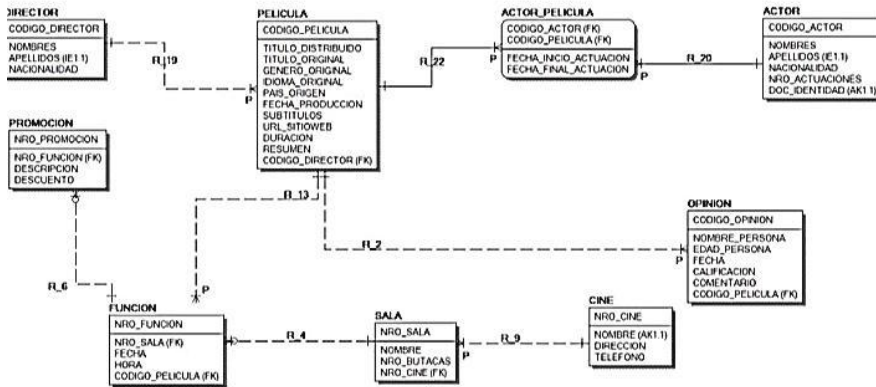
< Atrás   **Finalizar**   Cancelar   Ayuda

Clic en  
finalizar

> Resulta una nueva tabla



➤ Finalmente el modelo resulta de la siguiente forma:



Al cambiar el tipo de modelo (de lógico a físico) automáticamente las relaciones de muchos a muchos generan las nuevas tablas, sin necesidad de realizarlo manualmente.

## 5. Creando Áreas de Negocio y Stored Display

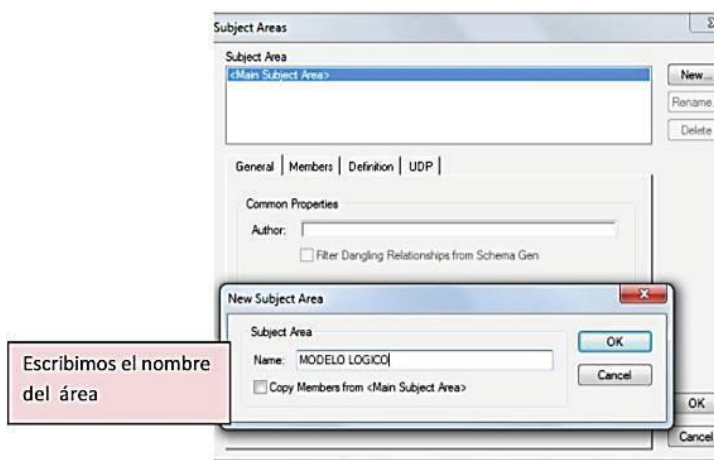
### 5.1. Áreas de Negocio.

Permiten trabajar con un subconjunto de entidades mientras se mantiene la integridad de todo el modelo.

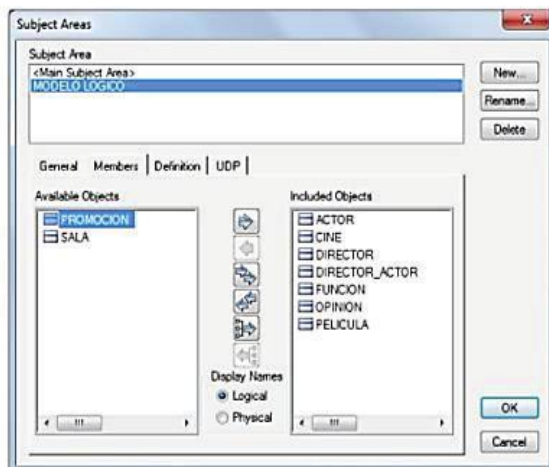
Para crear un Área de Negocio realizar lo siguiente:



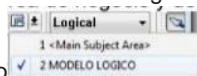
1. Haga clic en el icono de la barra de herramienta de ERwin
2. También desde el menú Edit elija Subject Área
3. ERwin le mostrara la siguiente ventana:
4. Clic en New.

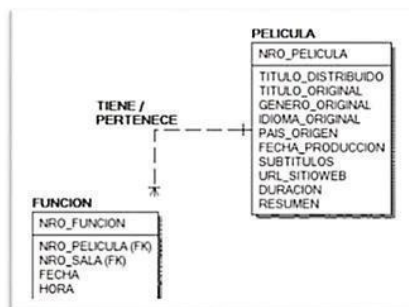


5. Clic en el TAB Members.



6. Seleccionar los miembros que participaran en el área de negocio.
7. Para visualizar realizar lo siguiente:
  - a. Seleccione la opción área de negocio y despliegue el botón de flecha hacia abajo
  - b. Seleccione el nombre del área de negocio que fue creada.
  - c. Finalmente se mostrara lo siguiente.

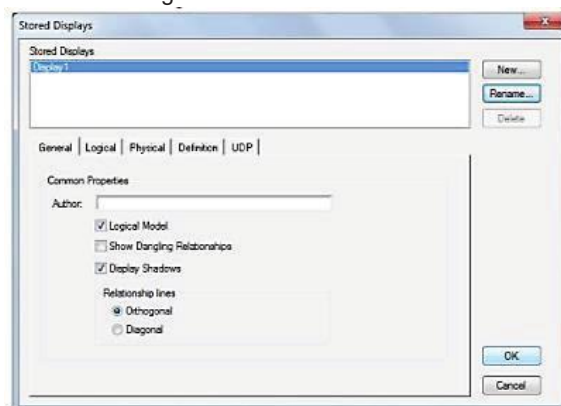




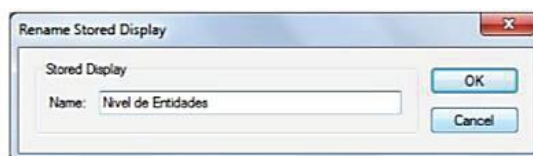
## 5.2. Stored Display

Permiten crear Tabs en la parte inferior de la ventana principal de ERwin. Para ello realizar lo siguiente:

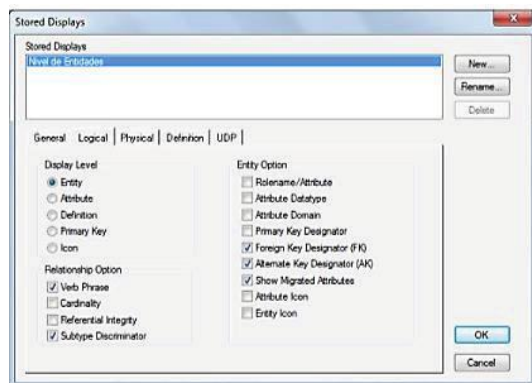
1. Desde el Menú Edit seleccione Stored Display...
2. ERwin le mostrara la siguiente ventana



3. Hacer clic en Rename, para cambiar el nombre

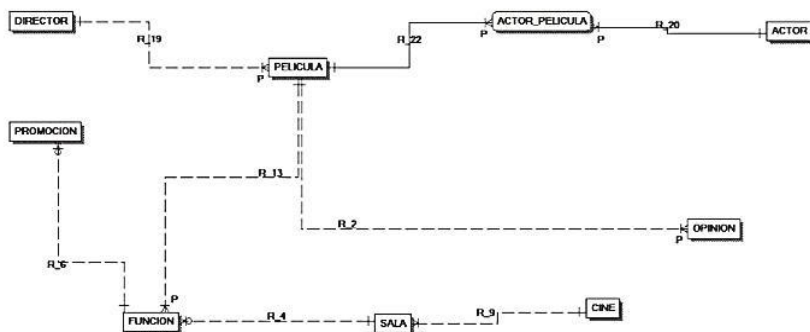


4. TABs para seleccionar el nivel apropiado de visualización y reordenar el modelo.



5. Clic en ok

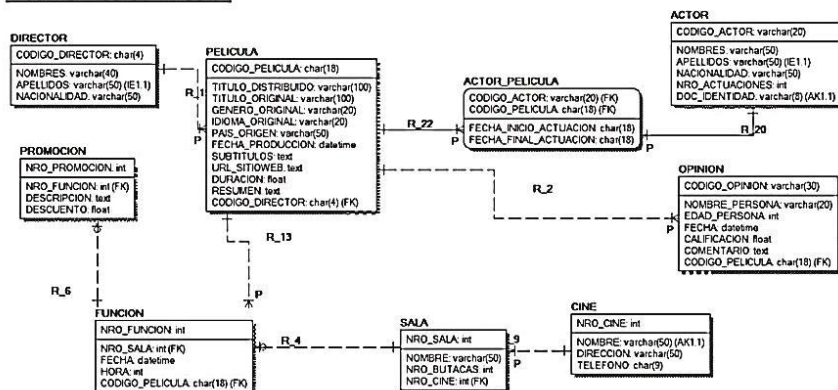
### NIVEL DE ENTIDADES



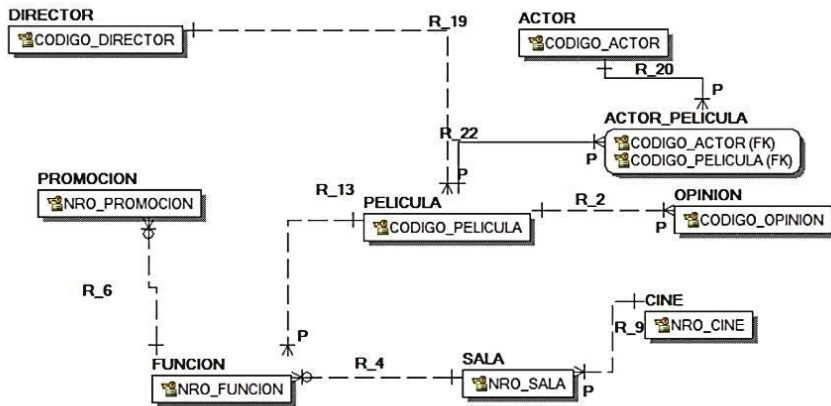
6. Para crear más Display clic en New.

7. Finalmente resultan las siguientes figuras.

### NIVEL DE ATRIBUTOS



### NIVEL DE LLAVES PRIMARIAS

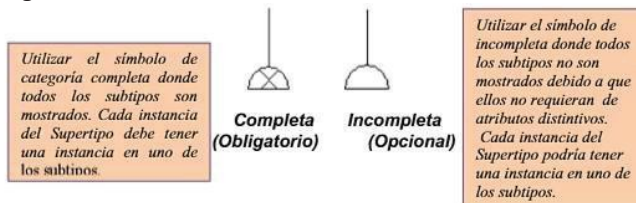


## 6. Optimizando el Modelo de datos.

### 6.1. Generalización Jerárquica.

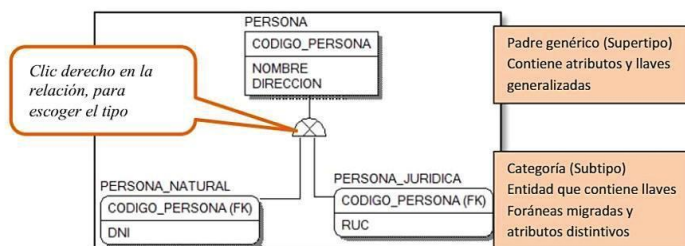
Se define como un conjunto de entidades que comparten atributos comunes.

**Simbología:**

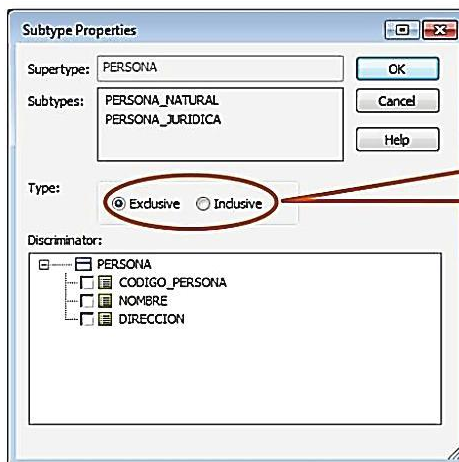


**Ejemplo:**

1. Hacer clic en



2. Seleccione la entidad Supertipo.
3. Seleccione la entidad subtipo (hijo)
4. Repita los siguientes pasos para cada entidad subtipo adicional a ser conectada.
  - a) Clic en el símbolo de relación de categoría.
  - b) Clic en la entidad subtipo



## 6.2. Integridad Referencial (RI)

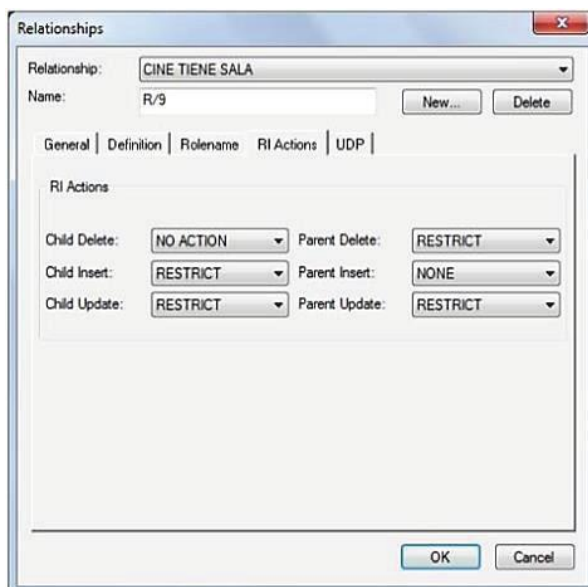
Se define como reglas que determinan lo que pasa cuando una fila de la entidad Padre o Hijo es insertada, actualizada o eliminada. Evita la data huérfana y asegura que las reglas de negocio se cumplan.

ERwin asocia por defecto opciones RI para cada acción en una relación, basado en el tipo de relación, acción y rol de la entidad (padre/hijo). Estos valores por defecto están basados en el sentido de GENERALIDAD; ellos pueden ser sobrescritos para todo el modelo o para una acción dada en una relación establecida.

- **Opciones de integridad referencial.**

Para cambiar las opciones de Integridad Referencial para un relación específica realizar lo siguiente:

1. Clic derecho en la línea de relación.
2. Elija la opción Relationship Editor...
3. En el editor de relaciones seleccione el TAB RI Actions
4. Seleccione las opciones de Integridad referencial desde el ComboBox
5. Clic en OK para retornar al modelo.

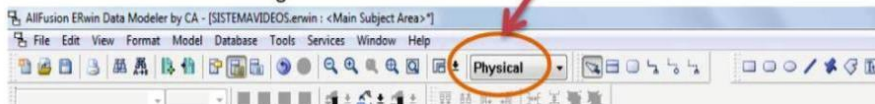


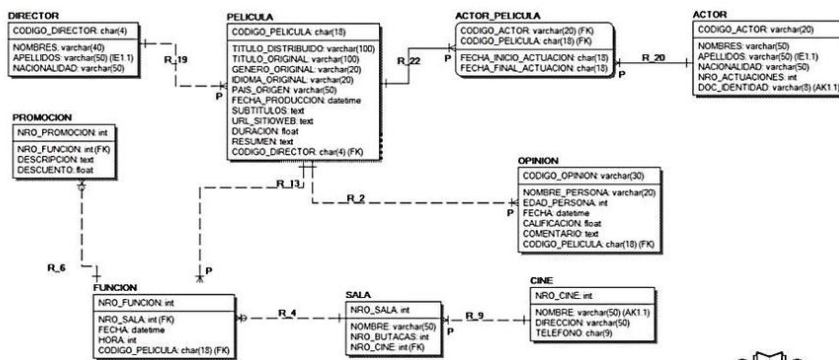
- ✓ Las relaciones de muchos a muchos solamente se muestran en el modelo lógico.
- ✓ Los símbolos de generalización jerárquica cambian en el modelo físico
- ✓ Entidades, atributos individuales o relaciones podrían ser diseñados como solo lógicos.

## 7. Modelamiento Físico.

ERwin mantiene las representaciones Lógica y Física del modelo de datos. Soporta características físicas del servidor objetivo (Target Server). Sincroniza el modelo físico de datos con el catalogo del servidor objetivo.

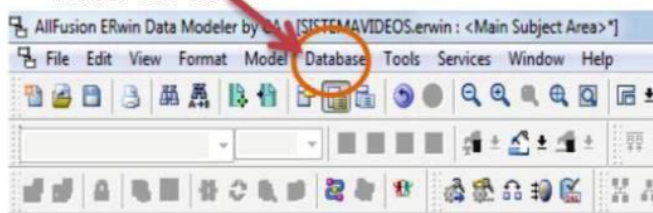
Para cambiar del modelo lógico al físico hacer clic en



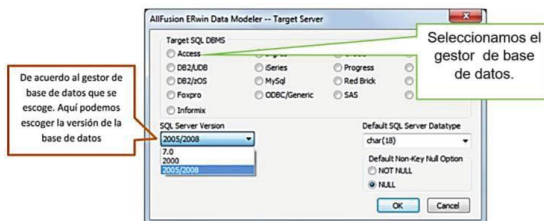


### 7.1. Seleccionar el servidor de la base de datos.

✓ Hacer clic en



Hacemos clic  
en choose  
Database



Para este ejemplo usaremos Acces.

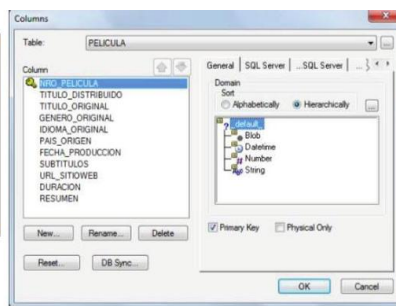
## 7.2. Propiedades de columna

Para asignar propiedades a las columnas realizar lo siguiente:

1. Clic derecho sobre la tabla
2. Elija Column

En esta ventana especifique la información del cliente y del servidor incluyendo tipo de dato y regla de validación.  
En el TAB General seleccione:

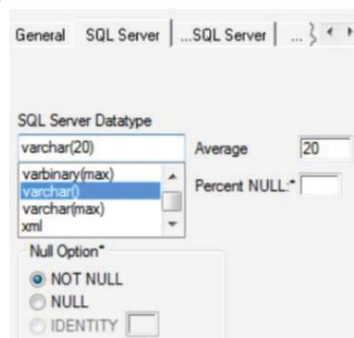
- Un dominio
- Configure la Llave Primaria
- Configure Physical Only



## 7.3. Asignación de tipos de datos

Para asignar tipo de dato a las columnas realizar lo siguiente:

1. Clic derecho sobre la tabla
2. Elija Column
3. Elija el TAB que tiene como nombre el servidor de base de datos elegido.



Elija el tipo de dato, Repita el mismo paso para cada dominio

## CUESTIONARIO

### Teórico:

1. Indicar las principales ventajas de usar Erwin.
2. Realice un cuadro comparativo de entre los tipos de llaves.
3. Describe las diferencias entre Modelo físico y lógico.

### Investigar:

4. Realizar un mapa mental sobre Microsoft Acces 2013.
5. ¿Qué es SQLserver?

### Practica:

6. Realiza el Modelo lógico (imagen N°1).

## Modelo de Facturacion

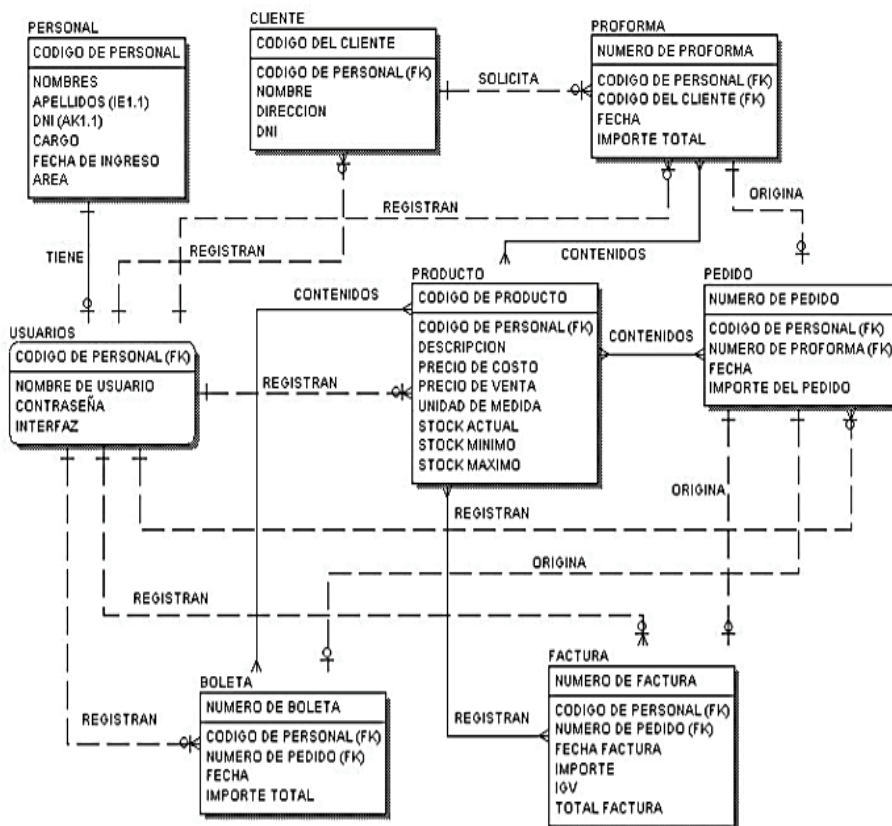


Imagen N°1



# ACTIVIDAD VIII

## **DISEÑO DE LA BASE DE DATOS RELACIONAL DE LA EMPRESA UTILIZANDO EL GESTOR DE BASE DE DATOS MICROSOFT ACCES 2013**

**OBJETIVO:** Diseñar la base de datos relacional ya normalizada en el sistema gestor de base de datos(SGBD); Microsoft Access 2013.

## INTRODUCCION AL DISEÑO FISICO DE LA BASE DE DATOS

### 1. Sistema Gestor De Base De Datos.

Los Sistemas Gestores de Bases de Datos (SGBD, por sus siglas en inglés), también conocidos como sistemas manejadores de bases de datos o DBMS (DataBase Management System), son un conjunto de **programas** que manejan todo **acceso a la base de datos**, con el objetivo de servir de **interfaz** entre ésta, el usuario y las aplicaciones utilizadas.

Gracias a este sistema de software específico el usuario puede **gestionar la base de datos** (almacenar, modificar y acceder a la información contenida en ésta) mediante el uso de distintas herramientas para su análisis, con las que puede realizar consultas y generar informes.

Además de gestionar los datos y mantener su consistencia, su utilización supone numerosas ventajas a la hora de construir y definir la base de datos a diferentes niveles de abstracción para distintas aplicaciones, pues facilita los procesos y también su mantenimiento.

### 2. Las funciones de un Sistema Gestor de Bases de Datos (SGBD)

La ejecución de las operaciones sobre la base de datos para luego proporcionarlos al usuario en función de su requerimiento se realiza de un modo eficiente y seguro. Sus características de un SGBD posibilitan el cumplimiento de una serie de **funciones**, que pueden agruparse de la siguiente manera:

- **Definición de los datos:** El SGBD ha de poder definir todos los objetos de la base de datos partiendo de definiciones en versión fuente para convertirlas en la versión objeto.
- **Manipulación de los datos:** El SGBD responde a las solicitudes del usuario para realizar operaciones de supresión, actualización, extracción, entre otras gestiones. El manejo de los datos ha de realizarse de forma rápida, según las peticiones realizadas por los usuarios, y permitir la modificación del esquema de la base de datos gracias a su independencia.
- **Seguridad e integridad de los datos:** Además de registrar el uso de las bases de datos, ante cualquier petición, también aplicará las medidas de seguridad e integridad de los datos (adopta medidas garantizar su validez) previamente definidas. Un SGBD debe garantizar su seguridad frente a ataques o simplemente impedir su acceso a usuarios no autorizados por cualquier razón.
- **Recuperación y restauración de los datos:** La recuperación y restauración de los datos ante un posible fallo es otra de las principales funciones de un SGBD. Su aplicación se realizará a través de un Plan de recuperación y restauración de los datos que sirva de respaldo.

### 3. Microsoft Access

Es un sistema de gestión de bases de datos Relacional creado por Microsoft (DBMS) para uso personal de pequeñas organizaciones.

Se ha ofrecido siempre como un componente de la suite Microsoft Office aunque no se incluye en el paquete “básico”.

Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas.

Entre las principales funcionalidades reseñables podemos indicar que:

- Permite crear tablas de datos indexadas.
- Modificar tablas de datos.
- Relaciones entre tablas (creación de bases de datos relacionales).
- Creación de consultas y vistas.
- Consultas referencias cruzadas.
- Consultas de acción (INSERT, DELETE, UPDATE).
- Formularios.
- Informes.
- Entorno de programación a través de VBA
- Llamadas a la API de windows.

### 3.1. Crear una base de datos de escritorio

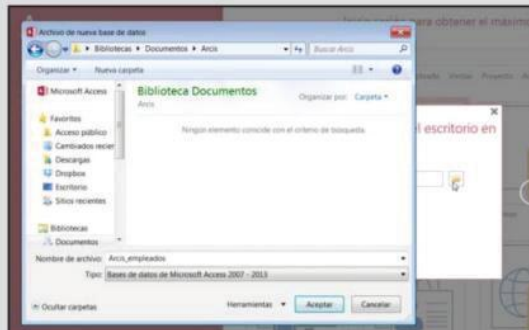
Cuando Access genera una base de datos desde cero, es decir, sin contenido predeterminado (a diferencia de las plantillas), tendremos que crear cada elemento que la integra de modo manual. En el siguiente Paso a paso aprenderemos a generar una base de datos de este tipo.



- 02** A continuación, ingrese el nombre que desee asignarle a la base de datos en la opción **Nombre de archivo**. Tenga en cuenta que no pueden existir dos bases de datos con el mismo nombre dentro de una misma carpeta.



- 03** Pulse el botón **Examinar...**, que se encuentra a la derecha del nombre asignado, para elegir dónde guardar la base de datos, ya sea en el disco rígido o en algún medio de almacenamiento transportable, como un pen drive.



04

Para finalizar, haga un clic sobre el botón **Crear**, ubicado debajo del nombre asignado a la base de datos.



Luego de cumplir con los pasos anteriores, se generará una tabla vacía en la nueva base de datos, que cerraremos por el momento, con el botón (X) situado en la esquina superior derecha de la pestaña de la tabla.

### 3.1.1. Tablas

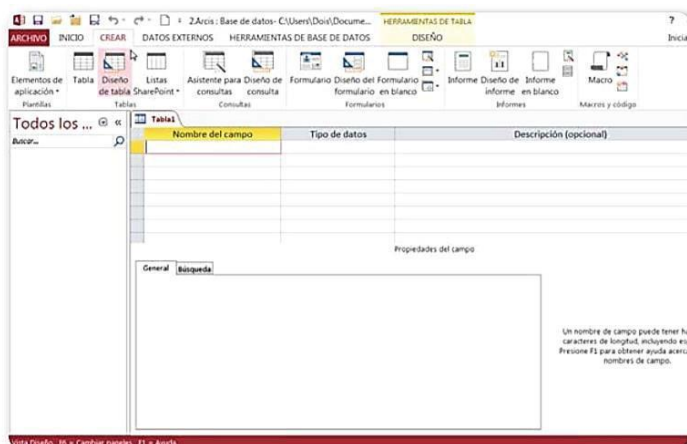
Las tablas (**relación**) son los elementos más importantes que integran una base de datos, ya que permiten ingresar información de manera ordenada y sistemática, además de buscarla y editarla posteriormente.

**Para crear tablas** debemos ingresar a la ficha:

1. Clic en Crear de la cinta de opciones y utilizar los botones del grupo Tablas.
2. Clic en la opción Diseño de tabla, además de generar una tabla, nos lleva a la **Vista Diseño**, que permite establecer y modificar su estructura: definir los campos que contendrá y el formato de los datos.

**Sus opciones principales son:**

- **Nombre del campo (atributos):** en esta sección tenemos que ingresar los nombres de cada uno de los campos que conforman la tabla, es decir, los nombres de las diferentes columnas.
- **Tipo de datos:** este apartado permite indicar la clase de datos (por ejemplo, número, texto, moneda, etcétera) que contendrá cada uno de los campos de la tabla. Debemos desplegar la lista de opciones y hacer clic sobre el tipo de dato elegido.
- **Descripción:** en este espacio podemos escribir comentarios sobre cada campo, es algo opcional.



### a) Tipos de datos

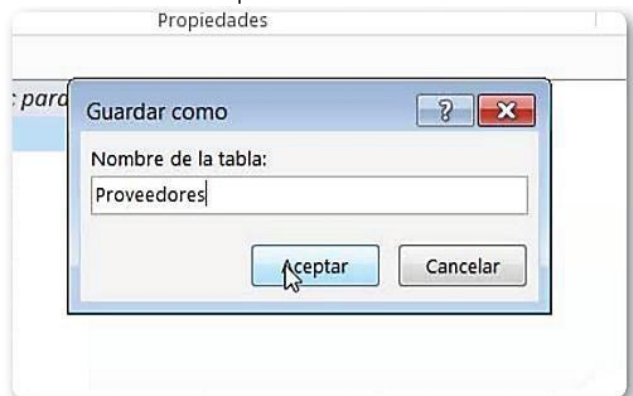
Cada registro (**tupla**) que introducimos en una tabla tiene un formato determinado, por eso Access permite definir el tipo de dato que se empleará en cada campo. Las opciones son las siguientes:

- **Texto corto:** permite registrar hasta 255 caracteres alfanuméricos, es decir que es posible ingresar texto y números que no sean utilizados para un cálculo, por ejemplo, números de teléfono y de documento.
- **Texto largo:** similar a la anterior, pero sin límite en la cantidad de caracteres.
- **Número:** solo admite números, que pueden ser usados en cálculos, por ejemplo, cantidad de horas trabajadas y cantidad de productos vendidos, entre otras posibilidades.
- **Fecha/Hora:** se utiliza para registrar fechas y horas, por ejemplo, las fechas de las ventas o los horarios de trabajo de los empleados.
- **Moneda:** representa valores monetarios, como, por ejemplo, salarios, precios de venta, costos, totales.
- **Autonumeración:** genera, de manera automática, números enteros correlativos o aleatorios.
- **Sí/No:** genera una casilla en la tabla, que se debe activar para indicar valores positivos o desactivar para valores negativos.
- **Objeto OLE:** para almacenar archivos, imágenes, sonidos y videos.
- **Hipervínculo:** vincula archivos y elementos de una base de datos a direcciones URL y de correo electrónico.
- **Datos adjuntos:** para adjuntar archivos a un campo, por ejemplo, una tabla que posea datos de los empleados puede llevar adjunto el currículum correspondiente, en el registro de cada uno de ellos.
- **Asistente para búsquedas:** permite generar un menú con opciones, para seleccionar los valores del campo. Este tipo de dato es para valores con poca o ninguna variación, por ejemplo, una lista de sucursales.

b) **Guardar y cerrar una tabla**

Después de finalizar el diseño de una tabla, debemos guardarla.

1. Para esto, pulsamos el botón **Guardar** de la barra de acceso rápido
2. En el cuadro de diálogo Guardar como, ingresamos un nombre para identificar la tabla y luego,
3. Pulsamos el botón Aceptar.



El nombre asignado se mostrará en la pestaña correspondiente. Si queremos cerrar la tabla, podemos hacerlo a través del botón Cerrar de la pestaña. Sin embargo, no lo haremos aún porque nos falta ingresar los registros. **La tabla guardada** aparecerá en el Panel de navegación, que se ubica a la izquierda de la ventana.

**3.1.2. Panel de navegación**

Este panel brinda la posibilidad de realizar operaciones complejas de forma rápida y dinámica, ya que permite acceder fácilmente a cada elemento de la base de datos de manera simultánea y llevar a cabo operaciones tales como copiar objetos, moverlos, eliminarlos, entre otras. Cada elemento que incorporemos en la base de datos se mostrará en este panel situado sobre la izquierda de la ventana. En la siguiente Guía visual analizaremos cada uno de sus elementos.

**01** **Menú de categorías:** mediante este menú podemos elegir la manera de visualizar los elementos de la base de datos según diferentes categorías, por ejemplo, por tipo de objeto, por fecha de creación o modificación, entre otras.

**02** **Ancho del panel:** muestra y oculta el panel.

**03** **Cuadro de búsqueda:** permite ingresar una palabra clave para filtrar los objetos mostrados. Esta opción es útil cuando la base de datos es extensa y en el panel no se visualizan todos los objetos que la integran.

**04** **Grupos:** muestran y ocultan los elementos que pertenecen al grupo.

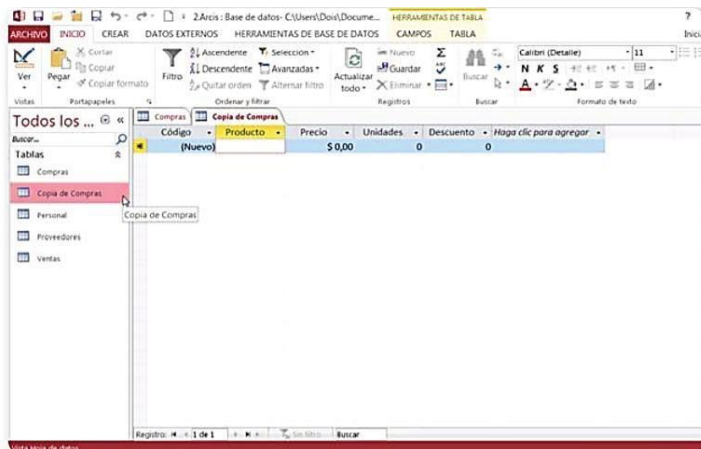
**05** **Objetos:** conjunto de todos los objetos que componen la base de datos: tablas, consultas, formularios, informes, macros, etcétera.

### Copiar objetos

Desde el Panel de navegación también es posible copiar cualquier objeto de una base de datos. Esta acción permite tanto duplicar únicamente la estructura de los elementos, como la estructura acompañada de sus datos almacenados. En este caso, **copiaremos una tabla**.

1. Seleccionamos en el panel y pulsamos el botón Copiar, de la ficha Inicio/ Portapapeles
2. luego, hacemos clic sobre el botón Pegar.
3. Se mostrará el cuadro de diálogo Pegar tabla como con las siguientes opciones:
  - **Nombre de la tabla:** aquí debemos indicar el nombre del duplicado.
  - **Opciones de pegado:** donde tenemos que elegir entre:
    - **Estructura solamente:** el duplicado tendrá el mismo diseño que el original, pero no poseerá registros.
    - **Estructura y datos:** el duplicado poseerá tanto el diseño como los registros contenidos en el objeto original.

- **Anexar datos a la tabla existente:** esta opción permite que los registros de la tabla original se copien en otra.
- A continuación pulsamos **Aceptar** y se creará la copia.



Para comprobar que la copia de la tabla es un duplicado, podemos mostrar su menú contextual:

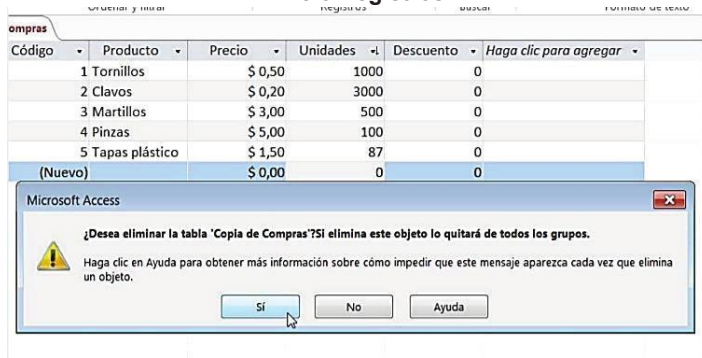
1. Hacemos clic con el botón secundario del mouse
2. Elegimos Vista Diseño.

Así, comprobamos que se muestran los mismos campos que en la tabla original. También podemos hacer doble clic sobre el panel para abrir la tabla y compararla.

### Eliminar objetos.

Esta es una de las operaciones que implican un mayor riesgo porque, al eliminar elementos, también borramos toda la información que ellos poseen. Para borrar un objeto de la base de datos:

1. Primero lo **cerramos**
2. Luego, lo seleccionamos en el Panel de navegación y pulsamos el botón Eliminar de la ficha **Inicio/Registros**.



### 3.1.3. Relaciones entre tablas.

Access 2013 es una de las herramientas más potentes para el manejo de datos porque, sin necesidad de procedimientos complejos, permite relacionarlos para obtener de ellos la información deseada en el momento en que la requerimos.

#### 3.1.3.1. Clave principal:

Para crear relaciones entre tablas, debemos indicar un campo especial en cada una de ellas, que se denomina clave principal.

En el proceso de elaboración de tablas, siempre debemos **incluir un campo** que nos permita identificar cada registro de manera única e irrepetible.

A continuación, en la Tabla 1, mostramos ejemplos de campos que identifican registros para distintas tablas.

| EJEMPLOS DE CAMPOS CLAVE |                                      |
|--------------------------|--------------------------------------|
| ▼ TABLA                  | ▼ CAMPO QUE IDENTIFICA CADA REGISTRO |
| Productos                | Código_artículo                      |
| Empleados                | ID Empleado                          |
| Sucursales               | Nº Sucursal                          |

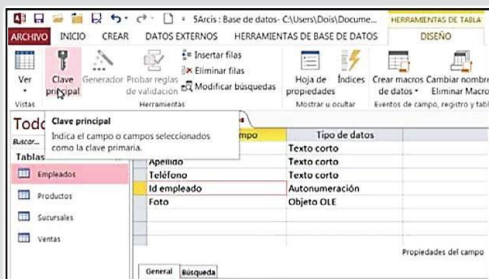
**Este método es fundamental** para localizar e identificar un dato, ya que permite hacer referencia a él sin nombrar cada uno de sus atributos.

**Por lo tanto**, es más simple mencionar al **empleado 1**, que decir Juan Pérez, 26 años, sexo masculino, número de documento, etcétera. El campo que se emplea para identificar cada registro con un valor único se denomina **campo clave o clave principal**. A su vez, el campo clave de una tabla es frecuentemente utilizado en otra, con el fin de poder relacionarlas entre sí. A estos campos los llamamos **clave externa**.

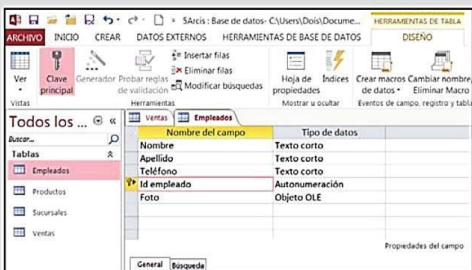
**Creación de claves principales** Por defecto, Access toma como clave principal el primer campo de la tabla, pero si queremos asignar uno diferente debemos seguir el siguiente Paso a paso.

## PAP: CREAR CLAVE PRINCIPAL

- 01** Ingrese en la Vista Diseño de la tabla, seleccione el campo al cual quiere asignarle la clave y en la ficha Herramientas de tabla/Diseño, pulse Clave principal.



- 02** Access mostrará un pequeño icono en forma de llave junto al campo, que indicará que es la clave principal de la tabla. De esta manera, podemos identificarlo en la Vista Diseño.



### Quitar y cambiar la clave principal

Como hemos visto, las tablas solo pueden contener un campo clave a la vez. Sin embargo, es posible que nos surja la necesidad de modificarlo o también de eliminarlo.

**Para cambiarlo**, simplemente debemos crear la clave principal sobre otro campo y de manera automática se borrará el actual, ya que, como mencionamos, no se permiten dos campos clave dentro de una tabla al mismo tiempo.

En el caso de que deseemos **quitar la clave principal** para que la tabla no contenga ningún campo clave, debemos ingresar:

1. En la **Vista Diseño de la tabla**
2. Seleccionar el campo que contiene la clave principal: **podemos** situar el cursor sobre una celda del campo o hacer clic sobre el cuadro gris que lo precede.
3. Luego, debemos desactivar el botón **Clave principal**, de la ficha Diseño o podemos hacer un clic derecho y utilizar el menú contextual. De esta manera, no se mostrará el icono en forma de llave y esto nos indicará que en la tabla no existen claves principales.

Al proceso de relación podemos definirlo como: la vinculación entre dos tablas a través de un campo en común. A continuación, veremos cómo es posible crear la relación entre tablas.

### 3.1.3.2. Tipos de relaciones

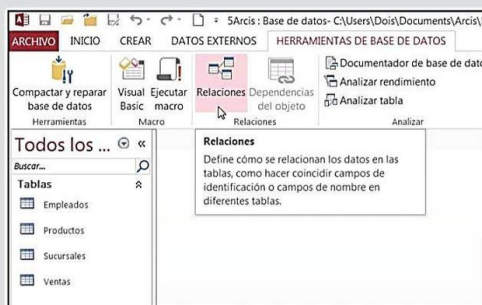
Los tipos de relaciones que se pueden establecer entre tablas son:

- **Uno a varios:** Los datos de un campo pueden tener múltiples registros en el campo de la otra tabla. **Por ejemplo**, un empleado puede tener registradas varias ventas, por lo tanto, el campo Id Empleado de la tabla Empleados posee relación uno a varios registros con respecto al campo Id Empleados de la tabla Ventas.
- **Varios a varios:** Los datos de un campo pueden tener varios registros en la otra tabla y viceversa. **Por ejemplo**, un artículo puede incluir varias ventas y, a su vez, varias ventas pueden referirse a un mismo producto.
- **Uno a uno:** en este tipo de relación, a cada registro de una tabla le corresponde un solo registro en la otra. **Por ejemplo**, el campo Id Empleados de la tabla Empleados tiene solamente un registro que le corresponde en el campo Id Empleado de una tabla que refleje el total de ventas de cada empleado, ya que no es posible que un mismo empleado tenga dos totales por sus ventas.

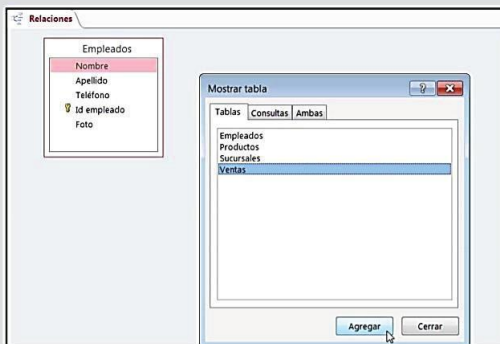
### 3.1.3.3. Proceso de relación

#### PAP: CREAR RELACIONES ENTRE TABLAS

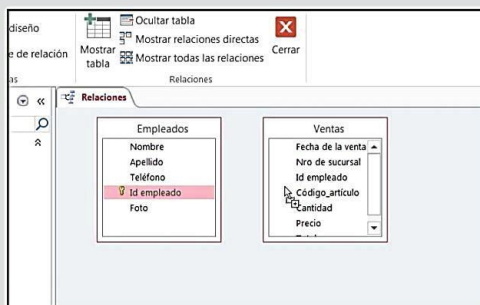
**01** Abra la base de datos que contiene las tablas que quiere relacionar. Ingrese en la ficha Herramientas de base de datos y active el botón Relaciones.



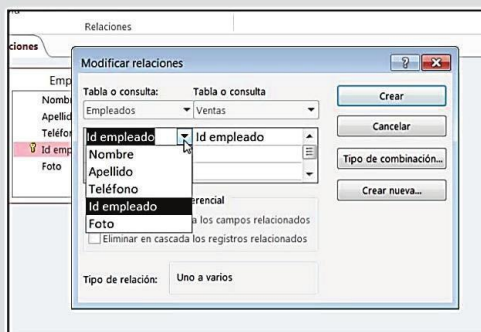
- 02** Se abrirá el cuadro **Mostrar tabla**, con el listado de tablas existentes. Seleccione las que quiere relacionar y pulse **Agregar**. Luego, presione **Cerrar**.



- 03** Arrastre el campo que ambas tablas tengan en común desde el cuadro de una tabla hacia la otra para relacionarlos.



- 04** En el cuadro **Modificar relaciones**, verifique que los campos sean los correctos. Para modificarlos, despliegue la lista y seleccione el deseado. Luego, pulse **Crear**.



**05** Se generará una línea que conecta ambas tablas a través del campo elegido. Pulse el botón Guardar, de la barra de acceso rápido.



**Al abrir una tabla** que está relacionada con otra, y ambas poseen el campo en común como campo clave, veremos la información correspondiente a la otra.

**Por ejemplo**, si el campo Id Empleado es clave principal en las dos tablas, cuando abramos una de ellas veremos la información correspondiente a la otra tabla y viceversa.

**En cambio**, si el campo en común sobre el cual se realiza la relación es campo clave en una de las tablas pero no en la otra, solamente veremos los datos de la otra al abrir la tabla que posee el campo clave. Es decir que la relación se da únicamente en un sentido.

**Por ejemplo**, si el campo Id Empleado es clave principal en la tabla Empleados pero no en la tabla Ventas, cuando abramos la tabla Empleados veremos los datos correspondientes a sus ventas y, al abrir la tabla Ventas, no veremos información sobre los empleados porque el campo Id Empleado no es clave principal en esta tabla.

**Luego de crear una relación**, es posible modificarla, para ello debemos hacer doble clic sobre la línea conectora entre las tablas. Recordemos que es posible realizar la cantidad de relaciones deseadas en una misma base de datos.

#### 3.1.3.4. Opciones para la relación entre tablas

En el proceso **de creación o modificación de las relaciones**, se muestra el cuadro de diálogo **Modificar relaciones**, que permite activar/ desactivar la casilla **Exigir integridad referencial**. Al activar esta casilla, se habilitan las opciones **Actualizar en cascada campos** relacionados y **Eliminar en cascada** los registros relacionados.



Al activar la casilla **Exigir integridad referencial**, indicamos que los datos entre ambas tablas deben ser coherentes: no se podrá eliminar un registro que contenga un dato relacionado.

**Por ejemplo**, todo valor que se ingrese en el campo **Id Empleado** de la tabla **Ventas** debe existir en el campo **Id Empleado** de la tabla **Empleados**. Si en la tabla **Empleado** se hace referencia al empleado 5, en la tabla **Ventas** debe existir ese registro.

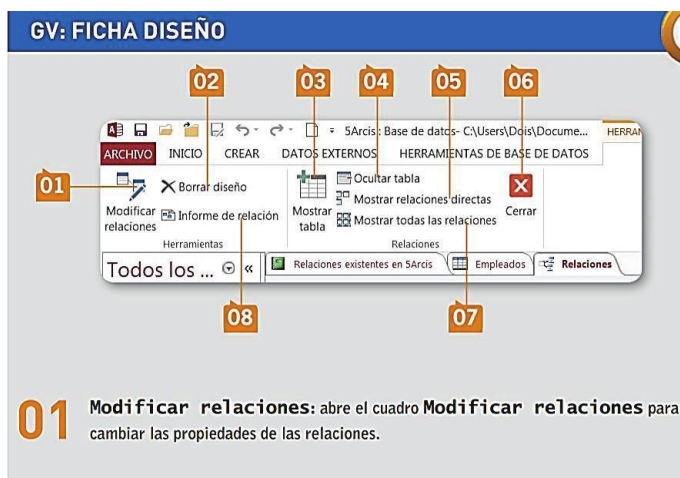
La casilla **Actualizar en cascada** campos relacionados permite actualizar de manera automática los cambios generados en uno de los campos que interviene en la relación.

**Por ejemplo**, si modificamos el número que identifica a un empleado en la tabla **Empleados**, este se actualizará también en la tabla **Ventas**. Si activamos la casilla **Eliminar en cascada** los registros relacionados, cuando borremos un registro que posee el campo clave, se eliminará automáticamente de las tablas que contengan este registro como clave externa. Por ejemplo, si borramos un empleado de la tabla **Empleados**, los registros que se correspondan en la tabla **Ventas** también se eliminarán.

### 3.1.3.5. Trabajar con relaciones

Una vez que hemos conocido los tipos de relaciones y las opciones de modificación que ofrecen, podemos trabajar con la manipulación del formato de los datos relacionados. Recordemos que relacionar tablas posibilita acceder de manera rápida a la información seleccionada de forma eficaz, por lo tanto, una adecuada edición de las relaciones es fundamental para aprovechar todo el potencial de dicha información.

**Ficha Diseño** En el momento en que podemos ver la pestaña **Relaciones**, es cuando aparece la ficha **Diseño** en la cinta de opciones. Esta ficha permite realizar varias operaciones sobre la relación, veamos para qué se utilizan sus botones.



- 02** **Borrar diseño:** elimina el diseño de la pestaña, sin borrar la vinculación establecida entre las tablas.
- 03** **Mostrar tabla:** abre el cuadro **Mostrar tabla**, que permite agregar tablas en la pestaña **Relaciones**, las cuales podrán ser utilizadas para nuevas relaciones.
- 04** **Ocultar tabla:** oculta la tabla activa, es decir, la tabla en la cual se encuentra un campo seleccionado. Es posible ocultar una a una todas las tablas.
- 05** **Mostrar relaciones directas:** permite visualizar solo las relaciones en las cuales participa la tabla seleccionada. Recomendamos ocultar todas las tablas, excepto la que queremos mostrar con sus relaciones directas y, luego, presionar este botón. Solamente se visualizarán las tablas que contienen relaciones directas con la tabla seleccionada.
- 06** **Cerrar:** cierra la pestaña **Relaciones**. Si existen cambios que no se guardaron hasta el momento, Access solicitará confirmación para almacenarlos.
- 07** **Mostrar todas las relaciones:** posibilita visualizar todas las relaciones. Si existen tablas ocultas que participen de relaciones, también se mostrarán.
- 08** **Informe de relación:** genera un informe con el diseño de la pestaña **Relaciones**, incluyendo el nombre de la base de datos y la fecha en la que fue creado.

### 3.1.3.6. Eliminar relaciones

Access 2013 permite eliminar las relaciones existentes a través de la **supresión** total de todo vínculo entre las tablas relacionadas. Para esto, debemos:

1. Ingresar en la ficha Herramientas de base de datos, presionar el botón **Relaciones** para que se presente la pestaña **Relaciones** y visualizar las relaciones existentes **en la base de datos**.
2. Luego, debemos hacer clic con el botón secundario del mouse sobre la **línea conectora** que representa la relación que deseamos borrar y en el menú, elegir la opción **Eliminar**.
3. Access mostrará un mensaje de confirmación, donde tenemos que pulsar el botón **Sí** para terminar el proceso y borrar la relación. Si pulsamos el botón **No**, no se realizará la eliminación. Después de

pulsar el botón **Sí**, se elimina la relación por completo, es decir que ya no existe vínculo alguno entre los datos de las tablas.

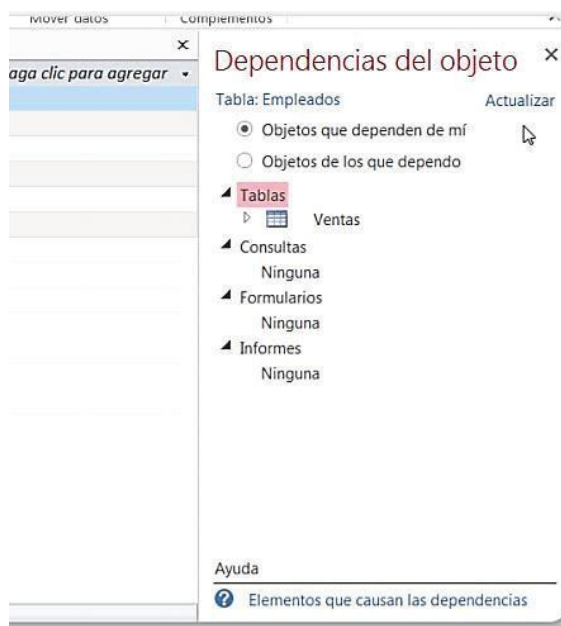
**Sin embargo, sus** ventanas siguen visibles aunque no exista la línea conectora entre ellas.

4. Si lo que deseamos es eliminar la pestaña de relaciones, es decir, borrar todas las ventanas de las tablas agregadas para relacionar, debemos pulsar el botón **Borrar diseño**, de la ficha **Diseño**.
5. Este **botón elimina el diseño** sin suprimir la vinculación entre las tablas. Luego de pulsarlo, Access mostrará un cuadro de diálogo que nos avisa que se eliminará la distribución de la pestaña **Relaciones**, y se quitarán todas las ventanas correspondientes a las tablas agregadas.
6. Si eliminamos el diseño de la pestaña, se **mostrará vacía, ocultará** todas las tablas y relaciones existentes. Si deseamos visualizarlas otra vez, podemos hacerlo mediante el botón **Mostrar todas las relaciones**, de la ficha **Diseño**, el cual presentará nuevamente todas las ventanas y relaciones existentes.

#### 3.1.3.7. Dependencias del objeto

El panel **Dependencias del objeto**, que nos muestra cuáles son todas las relaciones que tiene un objeto determinado. Para utilizarlo, primero seleccionamos el objeto en el Panel de navegación, y luego vamos a la ficha **Herramientas de base de datos/Relaciones** y pulsamos el comando correspondiente. **A continuación, el panel** aparecerá a la derecha de la ventana del programa. En este veremos el nombre del **objeto seleccionado** y luego **podremos elegir entre visualizar los elementos** que dependen de este o la relación inversa, es decir, de qué elementos depende el objeto en cuestión.

Más abajo se mostrarán las diferentes **categorías de elementos** con los que hay relación. Podríamos decir que este panel es una especie de mapa de relaciones, que nos muestra cómo está vinculado un determinado objeto.



Para cambiar el **objeto de origen**, debemos seleccionar uno diferente en el Panel de navegación y luego pulsar el vínculo **Actualizar en el panel Dependencias del objeto**. Veremos que se modificará el nombre del objeto en el panel y se mostrarán sus **relaciones específicas**.

Como podemos darnos cuenta, **el panel de dependencias** nos permite conocer, en apenas algunos minutos y unos pocos clics, cuáles son las relaciones que se han definido entre las diferentes tablas que integran una base de datos. Esta herramienta cobra aún mayor relevancia cuando trabajamos con muchos elementos y hemos establecido múltiples relaciones entre ellos.

#### 3.1.4. Insertar datos en la tabla.

El objetivo de las tablas es almacenar información, por lo tanto, uno de los procedimientos más importantes en la elaboración de una base de datos es la inserción de registros en las tablas. Es decir, ingresar la información que queremos almacenar en cada campo creado. Para esto, primero debemos visualizar la tabla en la **vista Hoja de datos**:

1. A la cual accedemos por medio del grupo **Vistas** de la ficha Inicio o a través de los iconos que se ubican en el sector derecho de la **Barra de estado**.
2. Hacemos doble clic sobre la **tabla** en el **Panel de navegación**. De esta manera, veremos la tabla con todos sus campos.  
En la hoja de datos (también llamada grilla) vamos a ingresar los diferentes registros que conformarán los campos de la tabla.

- Entonces, para introducir los datos, debemos hacer un clic con el mouse sobre la celda y escribirlos mediante el teclado. Para pasar de una columna a la otra podemos pulsar la tecla **TAB** o volver a hacer clic sobre la celda directamente.



### 3.1.5. Consultas

Mediante las consultas podemos seleccionar registros que cumplan una o varias condiciones, con el objetivo de utilizarlos para realizar operaciones como la visualización o la realización de cálculos. Llamaremos cálculo a toda operación que se efectúe por medio de operadores lógicos, tales como sumar, restar, multiplicar y dividir.

**Por ejemplo**, podemos calcular sueldos en base a horas trabajadas y precio por hora; la sintaxis sería: **[Horas trabajadas]\*[Precio por hora]**.

A través de distintos tipos de cálculos, obtenemos diferentes valores que no están escritos actualmente en las tablas, sino que se calculan. Sin embargo, debemos saber que no todas las consultas permiten realizar cálculos.

Existen distintos tipos de consultas que permiten seleccionar datos y se clasifican en: **consultas de selección** y **consultas de acción**.

### 3.1.6. Las consultas de selección

Son aquellas que posibilitan la elección de un conjunto de registros, con el objetivo de exhibirlos o disponer de ellos para futuras operaciones. Para esto, se establecen **criterios sobre los datos**, para que solo se presenten aquellos que los cumplan.

**Por ejemplo**, podemos indicar que se muestren el nombre y el código de los productos que cuesten más de \$390.

En este caso, tenemos un único criterio, que el precio sea mayor a 390. Sin embargo, es posible asignar más de uno.

**Por ejemplo**, podemos seleccionar aquellos productos que cuesten más de \$390 pero que se hayan vendido solamente en la sucursal 2.

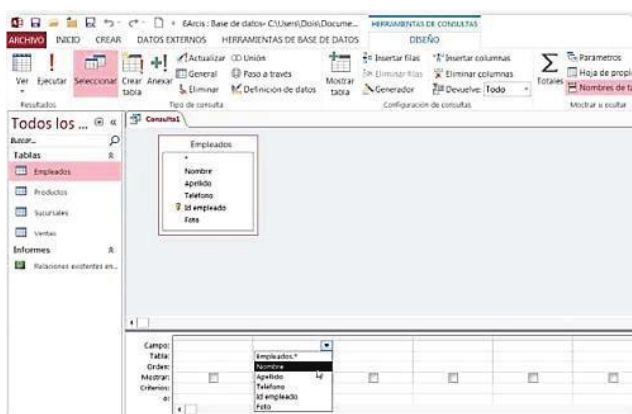
### 3.1.7. Diseño de consultas.

Las consultas se pueden crear manualmente o mediante un asistente. Para comenzar a trabajar con consultas, aprenderemos a hacerlo de modo manual, **a través de la Vista Diseño**.

Esta vista es la que permite establecer su estructura, es decir, indicar cuáles son las tablas que poseen los datos que vamos a seleccionar, así como también el o los criterios que utilizaremos para filtrar los datos.

**Para crear una consulta desde la Vista Diseño**, debemos:

1. Ingresar en la ficha **Crear** y hacer clic sobre el botón **Diseño de consulta**, del grupo Consultas.
2. A continuación, se presenta el cuadro de diálogo **Mostrar tabla**.
3. En este, debemos **seleccionar** todas aquellas tablas que contienen los datos que queremos consultar y tenemos que **pulsar el botón Agregar** cada vez. A medida que agregamos tablas, las vamos visualizando en la pestaña de la consulta.
4. Después de ingresar todas las tablas necesarias, pulsamos Cerrar.



### Agregar campos en las consultas

Después de agregar las tablas al diseño de la consulta, debemos indicar cuáles son los campos que participarán en la selección de datos. Es posible añadir desde uno hasta todos los campos de las tablas que hemos agregado.

En la **fila Campo** irán todos los campos que participarán de la consulta.

#### Ejemplo:

Seleccionar los nombres y apellidos de los **empleados** que tengan el número de identificación mayor a 6.

**Por lo tanto**, los que necesitaremos son: Nombre, Apellido e Id. **Empleado**, ya que son los únicos que contienen la información que deseamos seleccionar.

- **Para poner los campos en cada columna de la grilla:**  
Podemos desplegar la **lista Campo y seleccionarlos**, o hacer doble clic sobre el que queremos, en la ventana de la tabla que corresponda.  
Muchas veces, sucederá que el nombre de algún campo sea más ancho que la columna. Por eso, Access permite modificar

el ancho de las columnas de la grilla, ya sea de manera manual o automática.

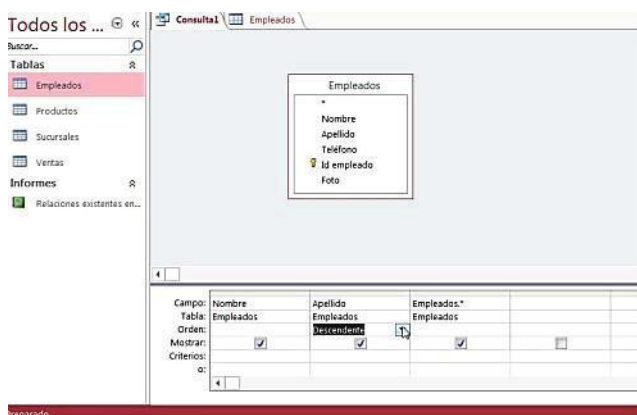
Para ajustarla manualmente, debemos:

- ✓ Hacer **un clic sostenido** desde el borde derecho superior hasta la nueva posición (ya sea para aumentar o reducir el tamaño).
- ✓ **Para ajustarla automáticamente**, debemos hacer doble clic sobre el borde y Access aumentará o disminuirá el ancho de la columna, dependiendo de su contenido.
- **La segunda fila de la grilla es la fila Tabla**, que muestra el **nombre** de la tabla que contiene el **campo** elegido en cada columna y no es posible modificarlo, ya que lo presenta de forma predeterminada.

**Para cambiarlo, habría que** modificar el nombre de la tabla directamente (la grilla muestra el nombre actual de la tabla). En este ejemplo, todos los campos derivan de la misma tabla.

- **Mediante la tercera fila de la grilla**, es decir la fila denominada **Orden**, podemos indicar que los datos resultantes de la consulta se ordenen de manera **ascendente o descendente**, de acuerdo con uno o varios campos. Debemos elegir el orden en la columna que contiene el campo con los valores que queremos ordenar.

**Para realizar esto, desplegamos** el menú de la celda correspondiente a la fila **Orden** y seleccionamos la opción elegida (Ascendente o Descendente). **Por ejemplo**, si queremos que la información correspondiente a los empleados que posean número de identificación mayor a 6 se muestren ordenados de forma descendente según su apellido, lo que debemos hacer es indicar el **orden en la columna que contiene el campo Apellido**.

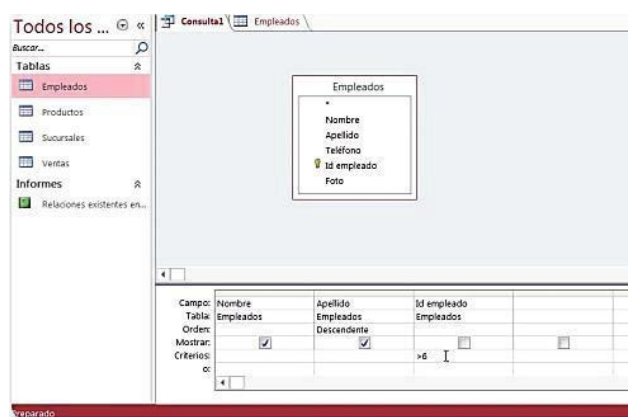


- **La fila Mostrar.** Esta fila posee una casilla de verificación en cada columna, que podemos activar o desactivar. Cuando la casilla correspondiente a un campo se encuentra activada, significa que dicho campo se mostrará en el resultado de la consulta. En nuestro ejemplo, vamos a desactivar la casilla correspondiente al campo Id. Empleado, ya que, aunque lo necesitemos en el diseño

para establecer el criterio, no deseamos visualizarlo en el resultado.

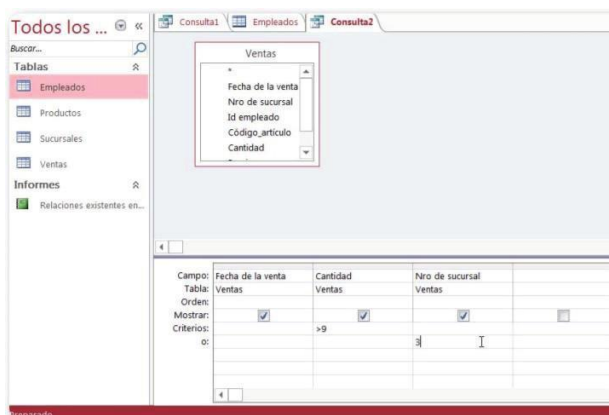
- **La fila Criterios;** el criterio de selección es uno de los campos más importantes de una consulta porque permite establecer una o varias **condiciones** que deben cumplir los registros para ser visualizados en el resultado de la consulta. Es decir que, cuando aplicamos criterios de selección, solo se mostrarán como resultado de la consulta aquellos registros que cumplan con las **condiciones especificadas**.

**Recordemos que en este ejemplo** queremos visualizar únicamente los datos de los empleados que tengan un número de identificación superior a 6, por lo tanto, el criterio de selección que debemos establecer es: >6.



- **Criterios opcionales** Access también ofrece la posibilidad de seleccionar registros que cumplan con una u otra condición, es decir, no es excluyente que ambas condiciones se den, bastará con que se respete, al menos, una de ellas.

**Por ejemplo,** un criterio puede ser el de visualizar las fechas de las ventas de la sucursal 3 y, otro, el de visualizar ventas que superen las 9 unidades. De esta manera, serán válidos tanto los registros de la sucursal 3 como los que superen las 9 unidades (sin importar a cuál sucursal pertenezcan). Cuando los criterios son opcionales, como en este caso, debemos escribir uno de ellos en la **fila O de la grilla**, para que la consulta no tome los dos en simultáneo.



- **Criterio de igualdad**

Por otro lado, los criterios también posibilitan seleccionar datos mediante la **comparación de igualdad**, que consiste en ver solo los registros que se identifican con un mismo criterio de consulta. Por ejemplo, si deseamos visualizar los registros correspondientes a la sucursal 1, debemos escribir en la fila **Criterios del campo** Sucursal el número 1. De esta forma, Access tomará como resultados válidos todos los registros que tengan el valor 1 en el campo Sucursal. Cuando la expresión de un criterio necesita comparación de **igualdad**, no hace falta escribir el carácter =, simplemente bastará con escribir el texto que debe ser igual en la tabla, para que se muestren los resultados en la fila Criterios del campo correspondiente.

- **Criterios de Fecha/Hora**

Quando los criterios deben estar en un campo de tipo **Fecha/Hora**, la fecha que se tomará como referencia debe escribirse entre signos numerales (#) para que Access la reconozca como tal. Por ejemplo, para visualizar las ventas realizadas después del 21 de noviembre de 2012, tenemos que escribir la siguiente expresión en la fila Criterios de la columna Fecha de la venta: >#21/11/2012#. En cambio, si queremos que los resultados muestren los registros de diferentes fechas, debemos incluirlas mediante la expresión In y separarlas por comas. De esta manera, Access reconocerá cada fecha por separado y podrá compararlas con los registros existentes en el campo de tipo Fecha/Hora de la tabla que contiene esos datos.

**Por ejemplo, para visualizar las ventas realizadas los días 23, 24 y 26 de noviembre de 2012**, tenemos que escribir la expresión: In(#23/11/2012#,#24/11/2012#,#26/11/2012#). El resultado de la consulta solo **mostrará las ventas de las fechas ingresadas**. También es posible que los criterios hagan referencia a datos establecidos entre dos valores. Para esto, debemos utilizar el operador Y, que permite incluir dentro de un mismo campo dos o más condiciones. Por ejemplo, para seleccionar las ventas en las que la cantidad de unidades vendidas **está entre 4 y 8**, tenemos

que escribir la siguiente expresión en la fila Criterios del campo Cantidad: >4 Y<8

### 3.1.8. Guardar consultas

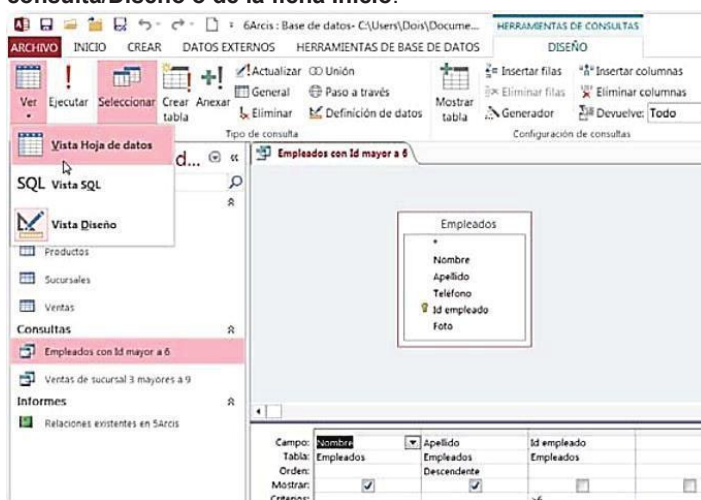
Para visualizar los resultados de una consulta, no es necesario guardarla previamente. Sin embargo, para conservarla y poder utilizarla en otro momento, sí debemos hacerlo. Para guardar una consulta:

1. Pulsamos **Guardar** en la barra de **acceso rápido**.
2. Aparecerá un cuadro de diálogo en el que debemos indicar el nombre que queremos asignarle a la consulta y luego pulsamos **Aceptar**.
3. Al guardar una consulta, esta aparecerá en el **Panel de navegación**, como sucede con todos los objetos que integran la base de datos.

### 3.1.9. Ejecución y resultados

Para visualizar los resultados de una **consulta**, debemos ejecutarla o ir a la **Vista Hoja de datos**. Todos los tipos de consulta se ejecutan del mismo modo, por esta razón los procedimientos analizados en esta sección serán válidos para todas las consultas.

- **Vista Hoja de datos** Al igual que las tablas, las consultas también poseen la **Vista Hoja de datos**, a la cual accedemos al desplegar las opciones del botón **Ver**, de la ficha contextual **Herramientas** de **consulta/Diseño** o de la ficha **Inicio**.

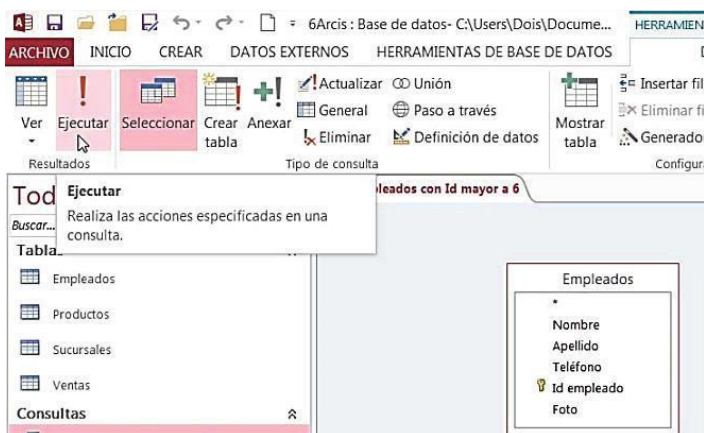


- **Formas de ejecutar consultas**

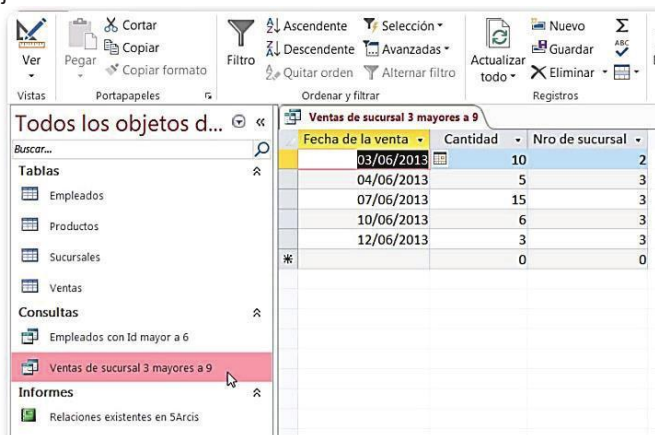
Todos los tipos de consulta pueden **ejecutarse** de dos modos: con la **cinta de opciones** y con el **Panel de navegación**.

Para ejecutar una consulta con la **cinta de opciones**:

1. Primero la consulta debe encontrarse visualizada en la **Vista Diseño**.
2. Luego, pulsamos el **botón Ejecutar**, situado en el grupo **Resultados** de la ficha **Diseño**, para aplicar todos los cálculos y acciones que hayamos configurado. En el caso de las consultas de selección, la ejecución solo mostrará los registros que cumplan con los criterios establecidos.



Para ejecutar una consulta desde el **Panel de navegación**, no es necesario que esta se encuentre abierta, pero sí **guardada** previamente; de lo contrario, no se mostrará el icono que la representa en el panel. Al hacer doble clic sobre el icono, la ejecutamos.



## CUESTIONARIO

### **Teórico:**

1. ¿Qué condiciones debe cumplir el contenido de un campo clave?
2. ¿Qué significa que un campo es clave en una tabla?
3. ¿Cómo se crea una relación entre dos tablas?
4. ¿Qué operaciones se pueden realizar desde el Panel de navegación?

### **Investigación.**

5. ¿Qué son los formularios en Access2013?
6. ¿A que llamamos consultas de acción, realice un ejemplo?

### **Práctico:**

7. En la base de datos Facturación: crear tablas, asignar claves y relaciones.
8. Realice tres ejemplos de consultas de selección con un criterio (use base de datos Facturación).

# REFERENCIAS

## **FUNDAMENTOS DE BASES DE DATOS**

Cuarta edición  
Abraham Silberschatz  
Henry F. Korth  
S. Sudarshan

## **INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS**

Escrito por C. J. Date ; Pearson Educación, 2001 - 936 páginas

## **MANUAL DE USUARIO ERWIN**

Marco A. Guevara  
Cesar R. Flores Nazario  
LibrosDigitales.net

## **SISTEMA DE INFORMACION**

Autores: Julián Pérez Porto y Ana Gardey. Publicado: 2008. Actualizado: 2008.  
Definicion.de: Definición de sistema de información  
(<https://definicion.de/sistema-de-informacion/>)

## **BASE DE DATOS**

<http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>  
Damián Pérez Valdés octubre 26/ 2007

## **MODELO ENTIDAD-RELACIÓN**

[https://es.wikipedia.org/wiki/Modelo\\_entidad-relaci%C3%B3n](https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n)

## **BASES DE DATOS**

[http://volaya.github.io/libro-sig/chapters/Bases\\_datos.html](http://volaya.github.io/libro-sig/chapters/Bases_datos.html)

## **ANÁLISIS Y DISEÑO DE BASE DE DATOS**

<https://es.scribd.com/doc/16271356/Informe-de-Base-de-Datos>  
**Jesus Osuna** el 27 de Mayo de 2014

## **INFORME DE BASE DE DATOS**

<https://es.scribd.com/document/40031583/Manual-Modelamiento-y-Diseno-de-Base-de-Datos-v0810>

## **MANUAL MODELAMIENTO Y DISEÑO DE BASE DE DATOS**

<https://es.slideshare.net/chelsin24/40031583anualmodelamientoydisenodebasededatosv0810>

## **ANÁLISIS Y DETERMINACIÓN DE REQUERIMIENTOS**

[www.dlsi.ua.es/people/jaime/apuntes/aesi\\_cap2.pdf](http://www.dlsi.ua.es/people/jaime/apuntes/aesi_cap2.pdf)

## **CARACTERÍSTICAS Y TIPOS DE BASES DE DATOS**

[https://www.ibm.com/developerworks/ssa/data/library/tipos\\_bases\\_de\\_datos/index.htm](https://www.ibm.com/developerworks/ssa/data/library/tipos_bases_de_datos/index.htm)  
Jorge Daniel Anguiano Morales

# REFERENCIAS

Consultor Certificado IM DB2 para LUW IBM México, 30-06-2014

## **HERRAMIENTAS CASE PARA DISEÑO DE BASE DE DATOS**

<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml#ixzz50j5ULHiS>

## **HISTORIA ERWIN**

<https://es.scribd.com/document/61158525/Historia-Erwin>

## **MODELAMIENTO DE DATOS UTILIZANDO ERWIN DATA MODELER**

Por: Aleksandr Paúl Quito Pérez.

## **MANUAL DE USUARIO DE ALLFUSION ERWIN DATA MODELER 7.2**

<http://es.calameo.com/read/00285969523004e2212b6>

Universidad Nacional José María Arguedas.

Ing. Richard Artemio Flores Condori.

## **ALLFUSION ERWIN DATA MODELER**

<https://oerntics2.wikispaces.com/file/view/Manual-de-Usuario-de-Erwin.pdf>

Marco A. Guevara Injoque

César R. Flores Nazario

## **Videos:**

Modelo entidad-relación

<https://www.youtube.com/watch?v=u2bXiPJf9oQ>

Importancia de las bases de datos elearning dominicanap

<https://www.youtube.com/watch?v=w9ueoFOY6gU>

Información e inteligencia para la toma de decisiones

<https://www.youtube.com/watch?v=9maeZ9slKwE>

Sistemas de información

[https://www.youtube.com/watch?v=82v3vfWV\\_sc](https://www.youtube.com/watch?v=82v3vfWV_sc)