

INTRODUCCIÓN A LAS BASES DE DATOS

Un poco de teoría

“Es un error capital teorizar antes de tener información”,
Arthur Conan Doyle, Sherlock Holmes, Escándalo en Bohemia.

TEMA A DESARROLLAR.....	2
CUESTIONES OPERATIVAS.....	7
TEMARIO	7
CONCEPTOS PRINCIPALES	7
LA INFORMACIÓN COMO RECURSO	8
UN POCO DE TEORÍA.....	10
1. HISTORIA.....	10
2. TÉRMINOS TÉCNICOS	12
CASOS	14
3. EL MODELO RELACIONAL DE INFORMACIÓN	15
3.1. Postulados.....	15
3.2. Relaciones entre tablas	18
3.3. Tipo de relaciones	20
3.4. Integridad Relacional	23
3.5. Vistas y Tablas	24
3.6. SQL - Structured Query Language	25
4. MODELO EAR (ENTIDAD-ATRIBUTO-RELACIÓN)	27
5. VENTAJAS Y DESVENTAJAS DEL MODELO RELACIONAL	33
5.1 Ventajas.....	33
5.2 Desventajas	33
EVALUACIÓN.....	34

NOTA:

Se aclara que se ha realizado un intento para no incluir nociones ni terminología que dificulten la lectura de esta primera clase, ya que el único objetivo pedagógico buscado es la introducción al tema.

Se considera que esta clase es fundamental para comprender el modelo que se utiliza en la actualidad para guardar información.

¿PARA QUÉ ESTUDIAR ESTE CURSO?

- Estamos rodeados de datos. Y diariamente se producen muchos más. Pensemos, por ejemplo, en la cantidad de información que se genera por cada compra con tarjeta de crédito, multiplicado por la cantidad de operaciones de ese tipo que se concretan por segundo en cada lugar de la Tierra. ¿Es mucho, verdad?
- Para almacenar y procesar todo este enorme caudal de información, se usan sistemas, gran parte de los cuales utilizan **bases de datos relacionales**.
- Por ese motivo, dominando esta técnica, tendrán una de las llaves de acceso al principal recurso estratégico del siglo XXI: **la información**.
- En la Era en la que vivimos, **la información es un recurso** y como cualquier recurso está protegido, tiene un costo, se compra y se vende, en función de su calidad.



TEMARIO



Curso

BASES DE DATOS PARA TODOS Prácticas con Microsoft Access

Certificado



Certificado de Asistencia emitido por la Facultad de Arquitectura, Diseño y Urbanismo de la Universidad de Buenos Aires

Profesor

Arq. Mercedes Frassia

Planificadora Urbana, FADU, Universidad de Buenos Aires Especialista en Sistemas de Información Geográfica, Manchester University, UK

Participantes

Ver listado Alumnos On Line

Objetivo

El objetivo principal del curso es introducir al asistente el manejo del programa Microsoft Access. Este curso va dirigido fundamentalmente a aquellas personas que necesitan organizar un sistema para almacenar y procesar un importante caudal de información.

Alcance

Curso teórico práctico. Cada clase esta compuesta por una parte teórica y un práctico. Se incluyen los conocimientos teóricos básicos antes de comenzar cada práctico. El curso se presenta con un estándar profesional adecuado para la formación laboral. Está diseñado para responder a las demandas de los gobiernos locales y nacionales, las organizaciones educativas, la industria y el comercio.

Clase 1

AGREGAR VALOR A LOS DATOS

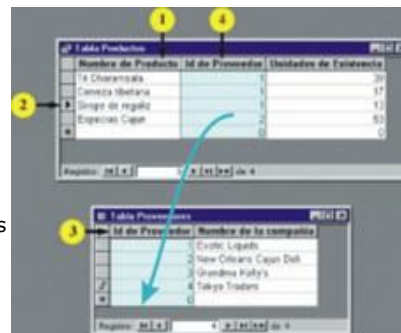
La información como recurso
Un poco de historia
Términos técnicos
El modelo relacional
Modelo EAR (entidad-atributo-relación)
Ventajas y desventajas del modelo



Clase 2

CREAR UNA BASE DE DATOS

Cómo crear una tabla
Cómo definir las características de un campo
Los distintos tipos de campo. Las propiedades de cada uno
Cómo importar tablas de otras aplicaciones
Relaciones entre tablas
Cómo establecer relaciones
Cómo garantizar la coherencia de los datos relacionados



CUESTIONES OPERATIVAS

Me presento

Tengo 52 años, soy arquitecta. He realizado un posgrado de especialización en Planificación Urbana en la Universidad de Buenos Aires y otro en Sistemas de Información Geográfica en UNIGÍS; Manchester, UK. Me estoy doctorando en la Universidad de Buenos Aires bajo la dirección de Michael Cohen, New School- NY y Nora Clichevsky, CONICET. Mi tema de tesis es "Información y Gobierno: la información hace la diferencia"

Hace años enseño Sistemas de Información Geográfica y Teledetección en cursos presenciales. Hace 3 años comencé a elaborar los cursos a distancia.

Mis áreas de interés son básicamente 2: la enseñanza y el desarrollo de software donde el GIS es una parte integral.

Tengo algunos desarrollos de los cuales me siento orgullosa como el Código de Planeamiento Urbano de la Ciudad de Buenos Aires, la base de edificios con Valor Patrimonial, y un software para la gestión de establecimientos agropecuarios. <http://www.cursogis.com.ar/Administrativa/mfrassia1.htm>

Pasemos al curso

El uso de la Web como medio posibilita una nueva forma de enseñanza en caminos que tanto alumnos y profesores debemos recorrer juntos. Yo personalmente he sido una alumna a distancia de la Universidad de Manchester, durante los dos últimos años, tarea que me requirió una dedicación semanal promedio de aproximadamente 15 horas, por lo cual, podría afirmar que mi experiencia como alumna es tan importante como la de docente.

Esta es la edición número 16 de este curso a distancia. En forma presencial lo di más de 20 veces, así que tengo una larga experiencia. Espero que lo disfrute.

La estructura del curso

El curso que usted está por comenzar, está estructurado en clases.

Cada una de las clases está diseñada respetando una unidad de sentido desde la perspectiva de la disciplina.

El tiempo estimado de trabajo por clase, incluyendo la lectura del material y las actividades, es de 4 hs. No obstante, es posible que estos tiempos varíen en forma significativa dependiendo de los ritmos y los conocimientos previos personales de cada alumno.

Al finalizar cada clase, usted deberá realizar una evaluación. La aprobación de esta evaluación es requisito para acceder a la próxima clase. Son muy sencillas, no se preocupe.

Le estoy enviando la primera clase. Para poder acceder a este archivo debe utilizar el programa WinZip. Este programa descomprime el archivo.

Si no lo tiene instalado en su PC puede bajarlo de <http://www.winzip.com/>

La versión de evaluación es gratuita.

Por otra parte, para leer la clase debe tener instalado Acrobat Reader, en su última versión. Si no puede leer el archivo que le mande, debe actualizar el programa bajándolo de <http://download.com.com/3000-2378-8338119.html?tag=lst-0-1>

Por supuesto, debe tener instalado el programa [Microsoft Access](#).

INSTRUCCIONES GENERALES DEL CURSO

No deje de visitar en este sitio el listado de Alumnos On Line

- Alumnos Matriculados:

Si usted es alumno matriculado, encontrará las instrucciones generales del curso en:

<http://www.cursogis.com.ar/Bases/Bases.htm>.

Pruebe los links a cada una de las clases.

Deberá solicitar su clave a info@cursogis.com.ar

- Alumnos NO Matriculados:

Si usted no es alumno matriculado, encontrará las instrucciones generales del curso en

<http://www.cursogis.com.ar/BasesP/BasesBienvenida.htm>

Si desea continuar con el curso debe inscribirse. Solicite mayor información a info@cursogis.com.ar

INSTRUCCIONES GENERALES DE LA CLASE

- Valido para todos los alumnos en general

Los recursos de esta clase los encontrará en el link

<http://www.cursogis.com.ar/BasesP/Bases1.htm>

Incluye la sección Evaluaciones.

CLASE 1



Clase 1



> Evaluación 1

> Evaluación 2

> Resolución Problema



1 Semana



TEMARIO: AGREGAR VALOR A LOS DATOS

La información como recurso

Un poco de historia

Términos técnicos

El modelo relacional

Modelo EAR (entidad-atributo-relación)

Ventajas y desventajas del modelo

¿Cuál es el modelo general que se utiliza actualmente para guardar información?

¿Qué es una base de datos?

¿Por qué todos los software de bases de datos declaran ser relacionales?

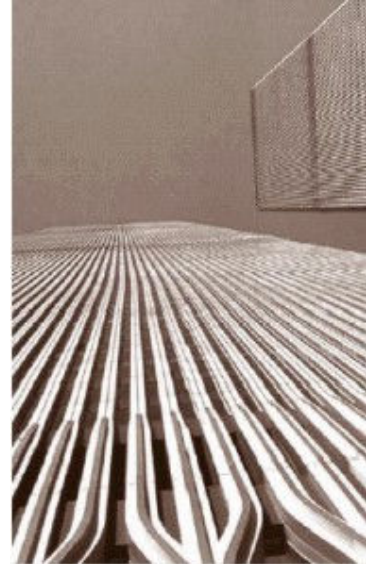
¿Cuáles son los términos técnicos que debo conocer?

¿Qué postuló Codd en un ya famoso artículo de 1970 a partir del cual se desarrollaron las Bases de Datos Relacionales?

¿Qué es un sistema de administración de bases de datos relacionales?

CONCEPTOS PRINCIPALES

- Imperan nuevas estrategias, procesos, estructuras organizacionales y tecnologías para asegurar que cada acción relacionada con la información este enfocada, coordinada y optimizada con el objetivo de agregar **valor a los datos**.



En esta clase encontrará respuestas a las siguientes preguntas:

- ¿Cuál es el modelo general que se utiliza actualmente para guardar información?
- ¿Qué es una base de datos?
- ¿Por qué todos los software de bases de datos declaran ser *relacionales*?
- ¿Cuáles son los términos técnicos que debo conocer?
- ¿Qué postuló Codd en un ya famoso artículo de 1970 a partir del cual se desarrollaron las Bases de Datos Relacionales?
- ¿Qué es un sistema de administración de bases de datos relacionales?

LA INFORMACIÓN COMO RECURSO

Los [datos](#) son observaciones que realizamos en el mundo real.

- Son recopilados como **hechos ó evidencias**
- Adquieren significado a **partir de su procesamiento**

DIFERENCIAS ENTRE DATOS E INFORMACIÓN

- Hay claras diferencias entre datos e [información](#), aunque generalmente se los confunde. Para tratar de entender la diferencia utilizaremos un ejemplo:

Dato: 36	Como es simplemente un número, no tiene en sí mismo ningún significado especial. Para que este valor sea de utilidad, es necesario interpretarlo en referencia a algún contexto particular. Como por ejemplo, especificar que ésta es la temperatura máxima registrada en la ciudad de Buenos Aires, en determinado año.
Información: 36°	Temperaturas máxima registradas en la ciudad de Buenos Aires, en el año 1994.

Datos: Los datos son hechos o cosas del mundo, lo suficientemente importantes como para ser registrados. Son recopilados como situaciones o evidencias. Adquieren significado a partir de su procesamiento para convertirse en información. Al hablar de datos, hacemos referencia a un concepto amplio que puede incluir texto, imágenes, sonido, reglas de decisión, etc.

Información: Es el resultado del análisis de los datos. Por ejemplo, un satélite colecciona datos (valores numéricos), que una vez volcados en un mapa, son interpretados por un experto que identifica los diferentes usos del suelo en determinada región.

- De este ejemplo podemos inferir que la **información es cualitativamente más que los datos** y esto es necesariamente así ya que la información debe servirle al usuario para tomar decisiones. Debe tener utilidad, condición que no siempre se cumple con los datos.
- Para disponer de información, se requiere de un proceso analítico sobre el dato.
- Al contar con **Datos** confiables y consistentes podemos generar **Información** que nos permita tomar las decisiones más convenientes.
- A partir de la información podemos generar **Conocimiento**, que no es más que: Información + Valor
- Hasta llegar al tope de la pirámide de valor, la **Sabiduría**, estatus en el cual contamos con la información valiosa en el momento, lugar y tiempo adecuados para la generación de ventajas competitivas.



- En las últimas décadas se han realizado importantes avances en la formalización de los modelos que definen la estructura organizativa de los datos, permitiendo su **almacenamiento, manipulación y consulta**.
- Estos avances han permitido llegar al concepto de **base de datos relacional**.
- Una **base de datos** es un conjunto de **información relacionada** que pertenece a una organización y que está **agrupada como un todo**. En la base de datos de una juguetería, por ejemplo, estará reunida la información de los juguetes (precio, cantidad en stock), así como los datos de los proveedores (dirección, teléfono, saldo deudor), clientes (si se desea llevar información individualizada de cada uno de ellos), empleados (salarios, presentismo, comisiones de los vendedores), contabilidad (cobranzas, pagos, liquidaciones), etc.
- Tener “agrupada como un todo” esta información trae muchos beneficios, como reducir la **redundancia y la inconsistencia de los datos guardados**.

UN POCO DE TEORÍA

1. HISTORIA

- En los 60, cuando las computadoras empezaron a desarrollarse, la atención estaba centrada en la resolución de problemas particulares: si era necesario procesar información, se programaba especialmente una aplicación particular que solucionaba la cuestión. Un ejemplo típico era la liquidación de sueldos de empleados: una tarea manual muy repetitiva que se automatiza para bajar el costo del procesamiento.
- Como el objetivo era la resolución de problemas, **se prestaba poca consideración al almacenamiento de datos**. Normalmente se archivaban en un formato específico (no-estándar) para cada aplicación. En otras palabras, cada programador trabajaba con un formato de datos propio.
- Este sistema fue adecuado mientras las aplicaciones permanecieron independientes.
- Los **problemas comenzaron cuando fue necesario compartir información** y los formatos que usaban las aplicaciones, no eran compatibles. Le doy un ejemplo: en una empresa había varias aplicaciones que guardaban datos en formatos diferentes. “Compras” mantenía la información sobre los artículos comprados; “Almacén”, sobre los que estaban en stock y “Ventas” sobre los artículos vendidos. Pero, por más que en los tres casos se hablara de los mismos datos (los artículos que la empresa comercializaba), no se podía compartir esa información, ya que las aplicaciones que había desarrollado cada sector, guardaba la información en un formato distinto que servía en forma específica para cada caso.
- Para solucionar este inconveniente, las organizaciones se veían obligadas a **exportar los datos** a los diferentes formatos que requería cada aplicación, tarea que era de por sí bastante engorrosa y no siempre posible.
- Pero el principal problema era que se debía **mantener varias copias** de la misma información en los diferentes formatos que requerían las aplicaciones existentes. Este procedimiento, inevitablemente creaba **redundancia de información**: imaginen en una empresa, la cantidad de veces que era necesario guardar el nombre de cada uno de los productos. A su vez, **traía aparejado inconsistencia**, ya que múltiples copias conllevan inevitablemente a errores. Por supuesto, el costo de mantenimiento era altísimo: si un cliente cambiaba de dirección había que cambiar el dato en tantos archivos como aplicaciones hubiera.
- Cuando la ciencia de la computación y la tecnología avanzaron, se extendió la necesidad de crear un sistema al cual accedieran todas las aplicaciones de la organización. Se pretendía, por ejemplo, que al ingresar los datos de un cliente, este registro sirviera a todos los sectores y que si este cliente cambiaba de domicilio, todas las aplicaciones lo registraran. Por supuesto, esta aspiración era absolutamente legítima e imperiosa.
- En definitiva, se buscaba que todas los sistemas en una organización (ventas, contabilidad, compras, etc.) **compartieran un sólo almacén de datos**.
- Es así como nace el **Database Management System (DBMS) o Sistema de Gestión de Base de Datos**, un sistema que servía de proveedor de datos a diversas aplicaciones.
- Este modelo fue propuesto originariamente en 1970 por un matemático de IBM, **Edward Codd**. Debe saber que aunque previamente habían surgido otros modelos (como el modelo jerárquico de datos, o el de red), fueron las “**Bases de Datos Relacionales**” las adoptadas expresamente por casi todos los sistemas comerciales conocidos.
- Gracias a su coherencia y facilidad de uso, el modelo se ha convertido a partir de los años 80 en el más empleado.

El modelo relacional aporta las **siguientes ventajas**:

- **Independencia lógica y física de los datos**, de las aplicaciones.
- **Redundancia mínima**, ya que la base funciona como repositorio común de datos para distintas aplicaciones.
- **Acceso concurrente por parte de múltiples usuarios**.

- **Distribución espacial de los datos**, ya que la independencia lógica y física facilita las bases de datos distribuidas. Los datos pueden encontrarse en otra habitación, otro edificio e incluso otro país. El usuario no tiene por qué preocuparse de la localización espacial de los datos a los que accede.
- **Integridad de los datos**, ya que se puede adoptar medidas de seguridad que impidan que se introduzcan datos erróneos. Esto puede suceder tanto por motivos físicos (defectos de hardware, actualización incompleta debido a causas externas), como de operación (introducción de datos incoherentes).
- **Consultas complejas por más de una condición.**
- **Seguridad de acceso y auditoria** Posibilita otorgar derecho de acceso diferenciado a personas y organismos. El sistema de auditoria mantiene el control de acceso a la base de datos, con el objeto de saber qué o quién realizó una determinada modificación y en qué momento.
- **Respaldo y recuperación**, que posibilita al sistema recuperar su estado en un momento previo a la pérdida de datos.
- **Acceso a través de lenguajes de programación estándar** que brinda la posibilidad ya mencionada de acceder a los datos mediante lenguajes de programación ajenos al sistema.

2. TÉRMINOS TÉCNICOS

Es importante que preste especial atención a esta sección, ya que introduce los términos técnicos a los cuales haremos referencia en la clase.

- **Database Management System (DBMS) o Sistema de Gestión de Base de Datos:** un sistema que sirve de proveedor de datos a diversas aplicaciones.
- **Base de datos:** Una base de datos es un conjunto de información relacionada con un asunto o con una finalidad.

Se compone de **entidades** (cosas u objetos del mundo real distinguibles de todos los demás objetos) que poseen **atributos** (propiedades o características de las que se quiere llevar registro).

Las entidades pueden ser cosas concretas, como personas o libros, o abstractas, como un préstamo o una venta.



Los atributos de la entidad "ALUMNO" podrían ser Documento de Identidad, nombre, apellido, edad, materias cursadas, notas obtenidas, etc.



Los atributos de la entidad "VENTA" podrían ser Fecha, Tipo de Factura, N° de Factura, Nombre del Vendedor, Nombre del Comprador, Artículos Comprados, etc.

- **Identificador o Llave:** Se llama identificador a todo atributo cuyo valor identifica la entidad bajo análisis en forma inequívoca. Si nos encontramos con una entidad que no tiene un atributo que lo identifique, podemos "inventar" un atributo que sirva como identificador.

¿Recuerdan haber realizado alguna vez una consulta en una biblioteca, donde se usaban fichas de cartón? Tiene presente que tenían un número con el que solicitaba el libro al bibliotecario: este número era el Identificador.



Si la entidad es "PERSONAS", su **Documento de Identidad** suele usarse como identificador



Si la entidad es "AULAS", se suele crear el atributo **Código de Aula**.



Si son "AUTOS", se usa la **patente**.



En un botánico que quiere tener identificada cada planta, **se crea un identificador**.

- **Query o consulta:** es una **declaración** o grupo de declaraciones en idioma formal que permite comunicarse con la base de datos **para recuperar la información almacenada**, especificando determinadas condiciones.
- **SQL (Structured Query Language):** SQL es un lenguaje para comunicarse con las bases de datos. Creado por IBM a mediados de los setenta. Oracle fue la primera empresa en utilizarlo en una base de datos. Actualmente existe un estándar SQL gestionado por un comité ANSI.
- **Inconsistencia de datos:** utilizaremos un ejemplo en lugar de una definición. Supongamos que una empresa guarda información de clientes. Si existen varios ingresos para cada cliente, se puede generar inconsistencia de información, ya que si este cliente cambia la dirección y nos olvidamos de

actualizar la misma en uno de los registros, al realizar una consulta, no se sabría cuál de las direcciones es la correcta.

CASOS

Antes de comenzar, vamos a plantearle 2 casos:

- El caso de una empresa. Utilizaremos los requerimientos de la empresa para explicarle el modelo relacional.
- Por otra parte le planteamos los requerimientos de una escuela primaria que requiere una base de datos para administrar sus operaciones. La resolución de este problema lo ayudará a adquirir el método necesario para crear una base de datos relacional. Por otra parte, podrá auto-evaluarse ya que a medida que vayamos avanzando, deberá resolver simples ejercicios cuya resolución podrá cotejar al final de cada unidad.
- Antes de continuar imprima estos casos.

EMPRESA

Las “reglas de la organización” de la empresa indican que:

- En la empresa hay varios **vendedores**.
- Que cada uno de los vendedores tiene un número de teléfono interno único y exclusivo.
- Que cada vendedor atiende varios **clientes**.
- Que cada cliente puede ser atendido sólo por el vendedor que le ha asignado la empresa.
- Que la empresa comercializa varios **productos**.
- Que cada cliente compra uno ó más productos que la empresa distribuye.

ESCUELA

Las “reglas de la organización” de la escuela indican que:

- En la escuela hay muchos **alumnos**
- Cada alumno tiene solo un **maestro**, pero un maestro tiene varios alumnos a cargo
- Cada alumno tiene un solo **responsable**, pero una persona puede ser responsable de más de un alumno a la vez. Un ejemplo típico, un padre con más de un hijo en la escuela.
- Cada responsable a cargo puede tener varios teléfonos. Un teléfono personal, varios laborales, un móvil.
- La escuela ofrece varias **actividades complementarias** optativas, como ser cerámica, francés e inglés. Un alumno puede hacer varias **actividades complementarias**. Obviamente, en cada actividad complementaria estarán inscriptos muchos alumnos.
- Por razones de privacidad, en una tabla aparte se desea tener la información acerca de la **salud del alumno**, como ser su grupo y factor sanguíneo, si tuvo apendicitis, si es alérgico a la penicilina, etc.

3. EL MODELO RELACIONAL DE INFORMACIÓN

- A medida que avanza encontrará una serie de preguntas que deberá ir contestando. Podrá cotejar las respuestas al final de esta sección, con el fin de auto-evaluarse.

3.1. Postulados

El modelo relacional establece los siguientes postulados:

- Una base de datos relacional siempre está compuesta por **tablas** que se conectan (relacionan) entre sí para compartir información.
- Una tabla es un conjunto de datos **sobre una entidad específica**: por ejemplo, en una base de datos de una empresa, se guarda en tablas diferentes información sobre los clientes, los vendedores y los productos.

CLIENTES			VENEDORES		

PRODUCTOS		

¿Qué tablas se requieren para administrar los registros de una escuela?

- Las tablas organizan la información en columnas llamadas **campos** y filas llamadas **registros**.

CAMPO 1	CAMPO 2	CAMPO 3

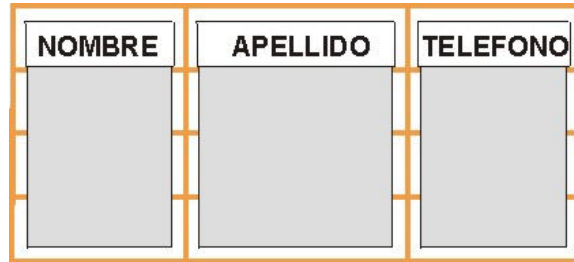
- Cada registro debe contener información de una **sola unidad de análisis**: por ejemplo, en la tabla "CLIENTE", una fila por cada cliente.
- Hay tantas filas como cliente tenga una empresa.

CLIENTE 1
CLIENTE 2
CLIENTE 3
CLIENTE 4

¿Cuántas filas tendrá la tabla ALUMNO?

- Los **campos** (columnas) **guardan los atributos** (propiedades o características) del registro.
- Los campos **pueden ser de distintos tipos**: numéricos (edad, cantidad de hijos), alfanuméricos (Nombre, Dirección), fechas (fecha de nacimiento, fecha de ingreso a la compañía), verdadero / falso (Posee auto propio, cumplió el servicio militar), etc.
- Todos los datos registrados en **una columna deben ser del mismo tipo**.

- El **nombre del campo debe ser único e inequívoco**: no se puede llamar a dos campos diferentes con el mismo nombre.
- Por ejemplo, si se quiere guardar 3 teléfonos por cada cliente, la primera columna se podría llamar Teléfonos_1, la segunda Teléfonos_2 y así sucesivamente.



- ¿Por qué no podemos guardar los 3 teléfonos en una sola columna, como haríamos en una ficha papel?
- Continué leyendo y descubrirá que, en el modelo relacional, las entradas deben ser indivisibles.
- Desarrollemos el ejemplo de una empresa. Analizaremos la información de los clientes.
- La empresa completa una ficha para cada cliente, especificando los datos personales del mismo.
- Esta información se podría guardar en la tabla "CLIENTE", que tendrá tantos registros (filas) como clientes tengan la organización.
- Cada registro se compondrá de un cierto número de campos (columnas), como ser N° de Documento, Nombre, Apellido, Dirección, Teléfono.

CLIENTE					
N° de Cliente	N° de Documento	Nombre	Apellido	Dirección	Teléfono
0023	10.551.985	Juana	Molina	Cabildo 110	4-784-6789
0056	11.346.897	Alberto	Alaluf	Lima 345	4-678-6782
0067	12.234.674	Maria	Jurez	Paz 897	3-457-8764
0098	12.456.432	Pedro	Colombo	Bolivia 877	6-567-8908

¿Qué campos serán necesarios para guardar la información de los alumnos?

- Así como cuando se organiza una biblioteca, se crea un índice que permite hacer una referencia a un libro fichado, en las bases de datos es necesario la creación de un campo denominado **identificador** que permita localizar la unidad de análisis. En este caso el cliente.
- En ningún momento dos filas de la misma tabla pueden tener el mismo identificador.
- En este caso, N° de Cliente es el atributo identificador que permite reconocer al comerciante en forma inequívoca.

¿Cuál será el identificador de la tabla MAESTRO?

- El modelo relacional introduce el concepto de **valor nulo**, para hacer alusión a valores que **no son todavía conocidos**. Por ejemplo, en la tabla clientes, la columna estatura puede estar vacía para determinado registro, no indicando esto que no tenga altura, sino que no se conoce la misma.

- En este ejemplo se ve claramente la diferencia entre “cero” y “nulo”.
- ¿Por qué el concepto de valor nulo es tan importante? Porque los registros con valores nulos **se excluyen de las estadísticas**.

Promedio estatura de clientes: 1.16 (un disparate)
Promedio considerando valores nulos: 1.74

NOMBRE	ESTATURA
CLIENTE 1	1.73
CLIENTE 2	
CLIENTE 3	1.84
CLIENTE 4	1.75
CLIENTE 5	
CLIENTE 6	1.63

¿En la tabla ALUMNO, qué diferencia habría entre alumnos que en el campo “alérgico a la penicilina” tuvieran las siguientes repuestas: no, si y nulo?

- **El orden de las columnas y las filas no tiene importancia:** Se podrá recuperar la información en el orden que se desee, sin importar la disposición en que fue ingresada.
- Las entradas de información **deben ser atómicas**. En otras palabras, debe ser información que no pueda ser divisible. Por ejemplo, en una ficha papel es admisible registrar dos idiomas (ingles, francés) en un mismo casillero, pero en el modelo relacional esto no es admisible. Una posible solución, aunque altamente desaconsejable, seria la creación de diferentes columnas: idioma 1, idioma 2, etc.
- ¿Por qué es desaconsejable? Porque si alguna persona no habla dos idiomas, estaremos desperdiciando espacio (creamos una columna que no tendrá datos). Y si habla más de dos... ¿Dónde lo escribimos? Este problema se denomina “**de tributos repetidos o de grupos repetitivos**”. Volveremos sobre este tema y daremos una solución más adelante en esta misma clase.
- ¿Recuerda el problema que se presentaba para guardar varios teléfonos por cliente? También es un típico ejemplo de atributos repetidos.

¿En la tabla ALUMNO y considerando que muchos alumnos, tienen más de un hermano, se podrían guardar en un mismo campo los nombres de todos los hermanos?

¿Se podría considerar a este, un problema de grupos repetitivos?

Antes de continuar, por favor verifique las respuestas a las preguntas que le hemos formulado.

Conéctese con <http://www.cursogis.com.ar/BasesP/Bases1.htm>

Las respuestas las encontrará en el link Evaluación 1

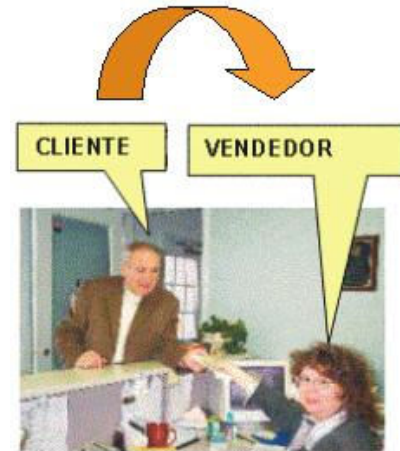
3.2. Relaciones entre tablas

Conteste las preguntas que le iremos haciendo a continuación.
Podrá cotejar sus respuestas al final de esta sección.

- Si ahora agregáramos en la tabla “CLIENTE”, al VENDEDOR que atiende a cada uno de ellos, con todos sus datos personales, surgiría una enorme cantidad de problemas. Además de tener que ingresar todos los datos del vendedor por cada cliente (**redundancia**), si algún empleado cambia de teléfono deberíamos actualizar la información en muchos registros, tantos como clientes tenga a cargo el vendedor, lo que llevaría indefectiblemente a generar **inconsistencias** (que en un registro figure el teléfono viejo y en otro el nuevo).
- También se agrandaría **el tamaño de la base**, lo que haría más costoso el medio de almacenamiento (discos rígidos) y más largas las búsquedas (porque se necesita pasar por registros más largos).
- Todos estos problemas se solucionan si se crea una tabla llamada “**VENDEDORES**”, donde cada registro corresponda a un empleado de la empresa del sector ventas.
- Nuevamente, nos debemos asegurar que cada registro tenga **su identificador único** (podría ser el legajo interno de la empresa).

VENDEDORES					
Nº de Vendedor	Nº de Documento	Nombre	Apellido	Sector	Interno
1	14.985.551	Juan	Aragón	Agroquímicos	23
2	11.897.346	Manuela	Sarquis	Fertilizantes	24

- Por otra parte, se debería **relacionar esta nueva tabla con la tabla CLIENTE**.



- Para relacionar la tabla VENDEDOR con la tabla CLIENTE, en la tabla CLIENTE creamos un campo donde se ubica el identificador del VENDEDOR que nos remitirá a los datos personales del mismo.
- A partir de esta relación, obtenemos para cada cliente, los datos del vendedor que le ha sido asignado.

VENDEDORES					
Nº de Vendedor	Nº de Documento	Nombre	Apellido	Sector	Interno
1	14.985.551	Juan	Aragón	Agroquímicos	23
2	11.346.897	Manuela	Sarquis	Fertilizantes	24

CLIENTES						
Nº de Cliente	Nº de Documento	Nombre	Apellido	Dirección	Teléfono	Nº de Vendedor
0023	10.551.985	Juana	Molina	Calles 110	4-784-8789	1
0056	11.346.897	Alberto	Alaruf	Lima 345	4-678-8782	1
0067	12.234.674	Maria	Jurez	Paz 897	3-457-8764	1
0098	12.456.432	Pedro	Colombo	Bolivia 877	6-567-8908	2

- Vamos a probar si funciona. ¿Sabe cuál es el nombre y apellido del vendedor que atiende al cliente 0023?
-
- Resumiendo, en general en una base de datos **hay más de una tabla**, que habitualmente están relacionadas entre ellas.
 - Las tablas se conectan a través de un campo común que ambas poseen: específicamente el **identificador** de una tabla puede ser **atributo de otra** con la finalidad de **proveer un vínculo entre ambas**.
 - Ese fue el caso de CLIENTE y VENDEDORES. El identificador primario de “CLIENTE” era Nº de Cliente, el identificador primario de “VENDEDORES” era el Nº de Vendedor y se relacionaban mediante el campo Nº de Vendedor.
 - En la tabla CLIENTE, al campo Nº de Vendedor, no es el identificador primario: **es el identificador secundario**.
-
- Recapitulando, se llama:
 - **Identificador Primario** (o llave primaria): El campo que identifica inequívocamente a una entidad.
 - **Identificador Secundario** (o llave foránea): El campo de una tabla que está vinculado con el identificador primario de otra tabla, creando así una relación.

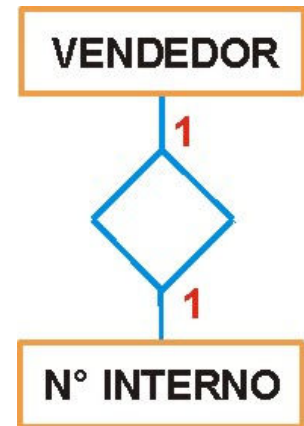
En el caso que le hemos planteado al principio de esta clase, se ha explicitado que cada maestro tiene a su cargo varios alumnos. ¿Piensa que convendría relacionar estas 2 Tablas? ¿Por qué?

3.3. Tipo de relaciones

Son varios los tipos de relaciones que se pueden entablar entre dos tablas. Los analizaremos a continuación.

UNO A UNO:

- Si cada unidad de la tabla A **está relacionada con una sola unidad de la tabla B y viceversa**, entonces la relación es Uno-a-Uno.
- En este ejemplo, cada vendedor tiene un sólo número telefónico interno asignado y cada número de interno es asignado a un sólo vendedor (la relación –rombo- se denominaría “*tiene asignado*” y es del tipo uno a uno).
- Resumiendo: cada vendedor tiene un número de interno exclusivo que no comparte.
- En este caso los **identificadores primarios coincidentes** permiten cruzar información entre tablas.



NOMBRE	IDENTIFICADOR	DIRECCION
	VENDEDOR 1	
	VENDEDOR 2	
	VENDEDOR 3	

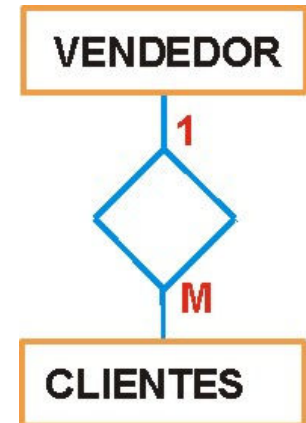
INTERNO	IDENTIFICADOR
101	VENDEDOR 1
102	VENDEDOR 2
103	VENDEDOR 3

- Por esta no es la única posibilidad de utilizar tablas cuyas relaciones es uno a uno. Este tipo de relación se utiliza frecuentemente para guardar **información diferenciando entre datos frecuentes, de ocasionales**. En el párrafo siguiente desarrollaremos este concepto utilizando un ejemplo.
- En una tabla de VENEDORES, se podrían archivar los datos usados frecuentemente, como: nombre, puesto, salario base, dirección, teléfono, fecha de nacimiento. En otra tabla, se podrían almacenar aquellos datos usados en forma excepcional como gustos musicales, pasatiempos, nombre, fecha de nacimiento y ocupación del cónyuge, nombres y fechas de nacimiento de los hijos, nombres de sus mascotas, auto que posee.

En el caso que le hemos planteado al principio de esta clase, se han explicitado las reglas de la organización. ¿Encuentra tablas que mantengan relaciones Uno-a-Uno? ¿Cuál?

UNO A MUCHOS:

- Si una unidad cualquiera de la tabla **A** **está relacionada con varias de la tabla B**, pero cada unidad de **B** **está en relación con sólo una unidad de A**, se dice que la relación es Uno-a-Muchos.
- En el ejemplo de la empresa, cada vendedor atiende a varios clientes, pero cada cliente sólo puede consultar al vendedor que se le ha asignado (la relación se podría llamar “*atiende*”).



- En este caso, las tablas se conectan a través de un campo común que ambas poseen: específicamente el **identificador** de una tabla puede ser **el atributo de otra**, con la finalidad de proveer un vínculo entre ambas.
- En este caso, el Identificador del vendedor será el Identificador Primario de la tabla VENDEDOR e identificador secundario de la tabla CLIENTE.

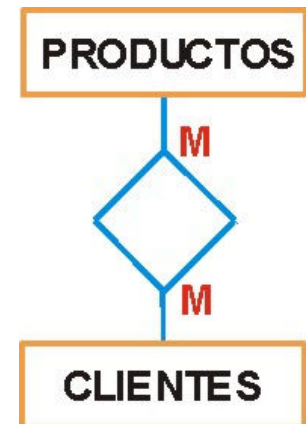
NOMBRE	VENDEDORES	DIRECCION
	VENDEDOR 1	
	VENDEDOR 2	
	VENDEDOR 3	

CLIENTES	VENDEDORES
CLIENTE 1	VENDEDOR 1
CLIENTE 2	VENDEDOR 1
CLIENTE 3	VENDEDOR 1

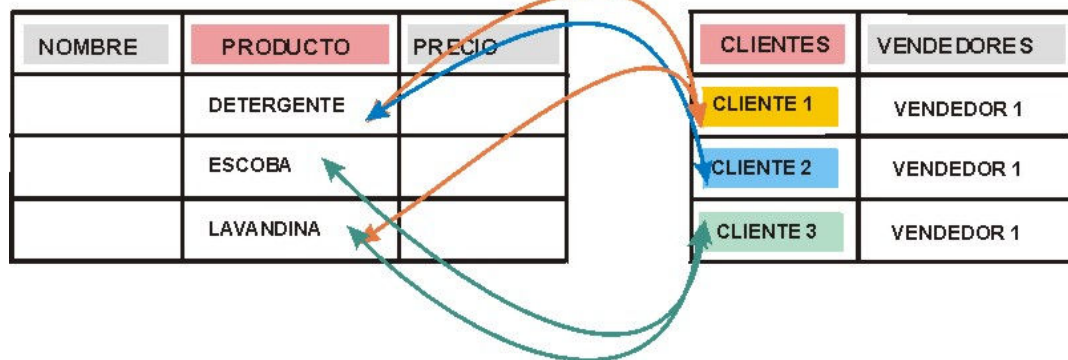
En el caso que le hemos planteado al principio de esta clase, se ha explicitado que para cada maestro hay varios alumnos. ¿Qué tipo de relación se debería establecer entre estas tablas? ¿Por qué? ¿Cuál será el campo a través del cual se relacionan?

MUCHOS A MUCHOS:

- Si una unidad cualquiera de la tabla **A**, **está relacionada con varias de la tabla B** y una unidad cualquiera de **B**, **está en relación con varias de la tabla A**, se dice que la relación es Muchos-a-Muchos.
- En el ejemplo de la empresa, son varios los clientes y varios los productos. Por este motivo, un cliente puede comprar varios productos, pero a su vez cada producto puede ser adquirido por varios clientes (se podría llamar a la relación “*compra*”).



- Por ejemplo, el cliente 1 compra detergente y lavandina; mientras que el cliente 2, sólo detergente; y el cliente 3, lavandina y escobas.
- En el modelo relacional, este tipo de relación **no está permitida**.
- Sin embargo, el modelo ofrece soluciones para estos casos, que son tan frecuentes en la realidad.



¿En el caso que le hemos planteado al principio de esta clase, puede descubrir que tablas mantendrían relación Muchos-a-Muchos? ¿Por qué?

Antes de continuar, por favor verifique las respuestas a las preguntas que le hemos formulado.

Conéctese con <http://www.cursogis.com.ar/BasesP/Bases1.htm>

Las respuestas las encontrará en el link Evaluación 2.

3.4. Integridad Relacional

- Ahora que ya conoce el funcionamiento de las claves primarias y las claves secundarias esta en condiciones de estudiar las **reglas de integridad**.
- Con este nombre se designa aquellas reglas que han de ser aplicadas a una base de datos para asegurar que los datos introducidos **sean consistentes con la realidad que se pretenden modelar**.
- Existen dos reglas *generales* que aporta el modelo relacional. Estas dos reglas son muy simples:

Primera Regla:

Regla de integridad de las entidades: ningún componente de la clave primaria de una relación base puede aceptar valores nulos.

- Esta regla impide la existencia de un registro sin identificador único.

Segunda Regla:

Regla de integridad referencial: la base de datos no debe contener valores de clave ajena o secundaria, sin concordancia con una clave principal.

- Esta regla impide que, por ejemplo, en una base de una escuela, exista un alumno con un responsable inexistente, o un responsable sin alumnos a cargo.
- Además, muchos RDBMSs añaden un buen número de características que ayudan a mantener más fácilmente la integridad de los datos. Mediante estos mecanismos es posible añadir reglas específicas para cada tabla en una base de datos; éstas son las denominadas *restricciones de integridad definidas por el usuario*. Por ejemplo, podríamos determinar que un alumno no pueda ser mayor de x años o que un arancel de un curso no pueda ser un valor negativo. El resultado sería que al intentar introducir un valor fuera de este rango, el DBMS rechazaría la información introducida mostrando un mensaje de error.

3.5. Vistas y Tablas

- En el modelo relacional se pueden distinguir distintos perfiles de usuarios. Por ejemplo: (Operador, Administrador, Gerente). A cada usuario, se le puede asignar un tipo distinto de permiso: sólo lectura, lectura y escritura pero sólo de ciertas tablas, etc...
- Con este objetivo, las tablas pueden ser de dos tipos: **tablas y vistas**.

• Las tablas son reales en el sentido que contienen la información .

• Las vistas son tablas virtuales . Se construyen a partir de tablas, siendo una combinación de filas y columnas de una o más tablas. El objetivo de las mismas es presentar la información a un particular usuario, en la forma que es útil para su propósito. Una vista puede ocultar los datos que el usuario no necesita o no debe ver.
--

- Supongamos que en un banco, hay una Base de Datos con las siguientes tablas:
CLIENTE (Nº de Cliente, Nombre, Apellido, Nº de Préstamo, etc.)
PRÉSTAMO (Nº de Préstamo, Nº de Sucursal, Monto, Cantidad Cuotas, etc.)
 - Un empleado a nivel gerencial necesita conocer toda la información disponible de los préstamos otorgados.
 - Sin embargo, los empleados no están autorizados a ver la información concerniente a los préstamos concretos que pueda tener cada cliente (monto, cuotas, etc.)
 - Por lo tanto, se le debe negar el acceso a la información fundamental de la tabla Préstamo. Pero se puede crear una Vista que muestre solamente si el cliente tiene o no tiene préstamos.
-

3.6. SQL - Structured Query Language

- El lenguaje de consulta de bases de datos relacionales por antonomasia, es como ya anticipábamos, el llamado **SQL** (Structured Query Language).

Las ventajas de este lenguaje son:

- Es **ampliamente usado** y, en general, todos los software comerciales lo reconocen.
 - Es un **idioma completo** que implementa todas las operaciones del modelo relacional.
 - Es un **idioma sencillo** con menos de 30 comandos.
 - El usuario **no necesita ser programador para poder usarlo**.
 - Adicionalmente a estas ventajas, **es el lenguaje utilizado cuando se quiere pasar información de un software a otro en bases de datos ubicadas en un sistema de red**. Claro ejemplo de esta habilidad es Microsoft ODBC (OPEN DATABASE CONNECTIVITY). Las bases de datos relacionales, a partir de los ODBC drivers, pueden intercambiar información.
 - **Esta implementado en ingles** y puede ser pensado como una frase estructurada en este idioma.
 - Las sentencias comienzan **siempre con un verbo** (SELECT, INSERT, UPDATE o DELETE) que le dice al software la acción a realizar. **Hay siempre un objeto**, usualmente una tabla, que es sujeto de la acción.
 - **La curva de aprendizaje de SQL es realmente rápida**.
-
- SQL se pronuncia deletreando en inglés las letras que lo componen, es decir "ese-cu-ele" y no "siquel" como se oye a menudo.
 - La historia de SQL empieza en 1974 con la definición, por parte de **Donald Chamberlin** del laboratorio de investigación de IBM, de un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional.
 - Este lenguaje se llamaba **SEQUEL** (**S**tructured **E**nglish **Q**uery **L**anguage) y se implementó en un prototipo llamado SEQUEL-XRM entre 1974 y 1975. Las experimentaciones con ese prototipo condujeron, entre 1976 y 1977, a una revisión del lenguaje (SEQUEL/2), que a partir de ese momento cambió de nombre por motivos legales, convirtiéndose en **SQL**.
 - El prototipo (System R), basado en este lenguaje, se adoptó y utilizó internamente en IBM y lo adoptaron algunos de sus clientes elegidos. Gracias al éxito de este sistema, que no estaba todavía comercializado, también otras compañías empezaron a desarrollar sus productos relacionales basados en SQL. A partir de 1981, IBM comenzó a entregar sus productos relacionales y en 1983 empezó a vender DB2.
 - En el curso de los años ochenta, numerosas compañías (por ejemplo **Oracle y Sybase**, sólo por citar algunos) **comercializaron productos basados en SQL, que se convierte en el estándar industrial**.
 - En 1986, **el ANSI adoptó SQL** (sustancialmente adoptó el dialecto SQL de IBM) como estándar para los lenguajes relacionales y en 1987 **se transformó en estándar ISO**. Esta versión del estándar va con el nombre de SQL/86. En los años siguientes, éste ha sufrido diversas revisiones que han conducido primero a la versión SQL/89 y posteriormente, a la actual SQL/92.
 - El hecho de tener un estándar definido por un lenguaje para bases de datos relacionales, **abre potencialmente el camino a la intercomunicabilidad entre todos los productos que se basan en él**.
 - Desde el punto de vista práctico, por desgracia las cosas fueron de otro modo. Efectivamente, en **general cada productor de software adopta e implementa sólo el corazón del lenguaje SQL**.

- Actualmente, está en marcha un proceso de revisión del lenguaje por parte de los comités ANSI e ISO, que debería terminar en la definición de lo que en este momento se conoce como SQL3. **Las características principales de esta nueva encarnación de SQL deberían ser su transformación en un lenguaje stand-alone** (mientras ahora se usa como lenguaje hospedado en otros lenguajes) **y la introducción de nuevos tipos de datos más complejos que permitan, por ejemplo, el tratamiento de datos multimediales.**

- Microsoft Access, implementa en forma interna el SQL. Lo traduce y reacomoda para facilitar la tarea. Obtiene de esta forma, una interfase amigable y sencilla de usar. Pero de fondo, esta siempre SQL.

4. MODELO EAR (ENTIDAD-ATRIBUTO-RELACIÓN)

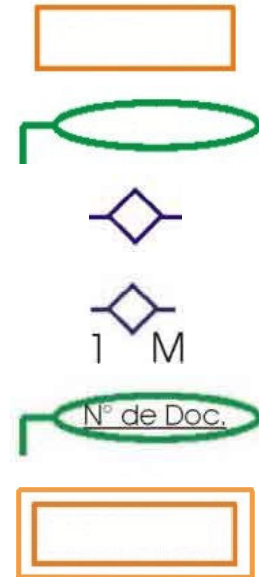
- Vamos a introducir a continuación un modelo que le facilitará la construcción de la base de datos.
- Este modelo fue creado en 1976 por **Peter Chen** del "Massachusetts Institute of Technology" y es ampliamente usado como un medio de modelado de bases de datos.

- En el modelo Entidad/Relación, la realidad se representa mediante un número muy reducido de conceptos semánticos básicos, que detallaremos a continuación.

- El mundo está compuesto de **entidades**. Una entidad es cualquier objeto distinguible y relevante en el mundo en cuestión (los clientes de nuestro ejemplo anterior).
- Las entidades poseen un número indeterminado de **propiedades**, que son "trozos" de información que describen a las entidades de uno u otro modo.
- Cada una de las entidades tiene un identificador, en otras palabras **son identificables de forma única e inequívoca**.
- Grupos de entidades conectadas mantienen **relaciones** con otros grupos de entidades.
- Se supone que mediante estos simples componentes se puede modelar cualquier "sección de realidad".

- El modelo EAR aporta una herramienta de modelado para representar las entidades, propiedades y relaciones: **los diagramas Entidad/Relación**.
- Mediante éstos, el esquema conceptual abstracto puede ser **graficado**, logrando mantener una independencia conceptual con respecto a la implementación propiamente dicha.
- En realidad, podemos hacer que los diagramas sean un fiel reflejo de las relaciones, interrelaciones y atributos del modelo relacional de datos.

- En estos diagramas, las **entidades** se dibujan como **rectángulos**
- Sus **atributos**, como **elipses** atadas a los rectángulos por líneas
- Las **relaciones** como **rombos**
- El **tipo de relación entre dos entidades** se representa mediante la convención que se muestran en la imagen: Uno-a-Muchos en este ejemplo.
- Una propiedad cuyo nombre **está subrayado**, es la **clave primaria** ó identificador.
- Finalmente, **un rectángulo doble**, significa que esa entidad es dependiente o débil, es decir, su existencia depende de la existencia de otra entidad. Por ejemplo, se podría establecer la restricción de que un teléfono no pudiera existir sin estar adscrito a un cliente.



Mientras resolvemos juntos el caso de la empresa, usted cree un diagrama Entidad-Atributo-Relación para el caso de la escuela.

Analice el problema paso a paso. Encuentre una respuesta en cada sección antes de seguir avanzando.

Podrá cotejar su respuesta al final de este apartado.

1^{er} PASO: IDENTIFICAR LAS ENTIDADES

- Las entidades son objetos o cosas que pueden identificarse y tener una existencia independiente y sobre las cuales la organización necesita información.
- Son en general **sustantivos**. En nuestro ejemplo, luego de un primer tanteo en la empresa, encontramos las siguientes entidades: los CLIENTE, cada uno de los cuales es atendido por un VENDEDOR; cada vendedor tiene un N° de teléfono INTERNO; los clientes compraban diversos PRODUCTOS.
- Cada una de las entidades es representada con **un rectángulo** que contienen el nombre de la entidad.

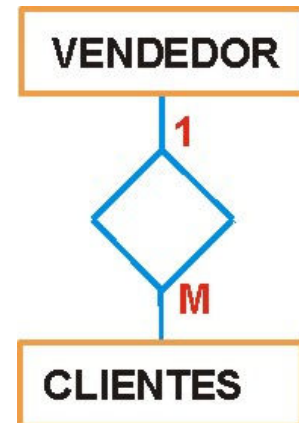
CLIENTES

- En este ejemplo hay 4 entidades: CLIENTE, VENDEDOR, INTERNO, PRODUCTO
- Por cada entidad se crea una tabla.

¿Cuántas entidades puede detectar en el ejemplo de la escuela? ¿Cuántas tablas necesitará crear?

2^{do} PASO: IDENTIFICAR LAS RELACIONES ENTRE TABLAS ESPECIFICANDO TIPO DE RELACIÓN

- Una relación es **una asociación entre dos objetos**. Las relaciones son representadas por **un rombo** entre los rectángulos. A veces se coloca el nombre de la relación dentro del rombo.
- En este momento es importante **reconocer los tipos de relaciones**. Las posibilidades son:
 - Uno-a-Uno
 - Uno-a-Muchos
 - Muchos-a-Muchos



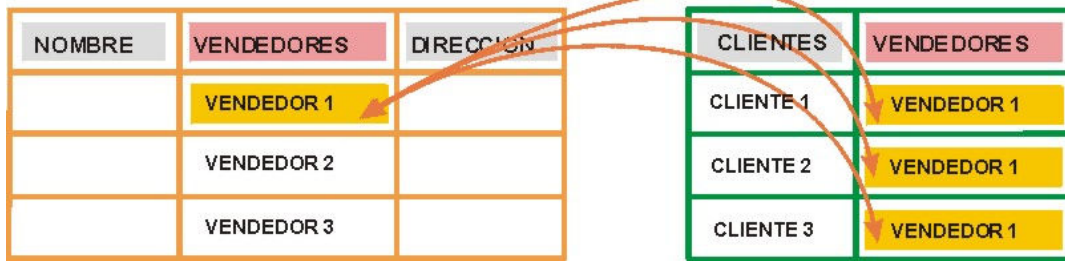
- Afortunadamente, las reglas para transformar las relaciones del diagrama, en relaciones del modelo relacional, son bastante directas y simples.
- Vamos a analizarlas a continuación.

Uno-a-Uno:

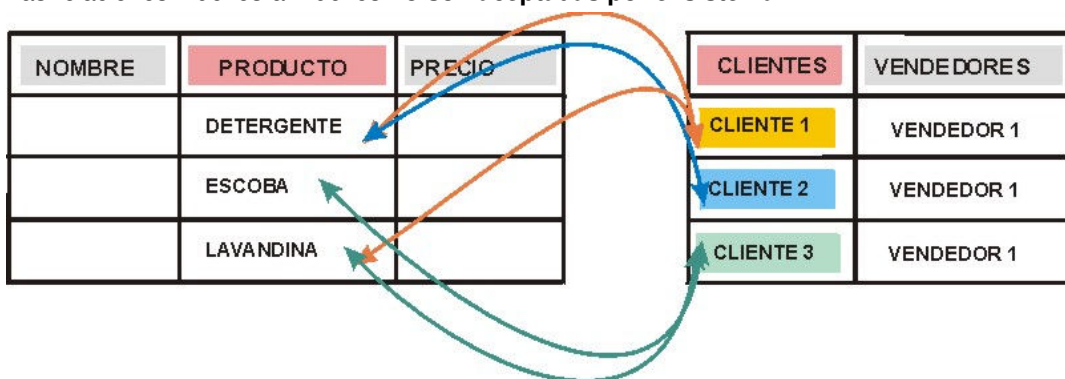
- En el ejemplo de la empresa, VENDEDOR e INTERNO mantienen relaciones Uno-a-Uno.
- Si la relación es Uno-a-Uno, es aceptable **unir estas entidades** en una sola tabla.
- En este caso se podría considerar al N° de Teléfono interno como un atributo del vendedor.
- Alternativamente, **podría ser más conveniente guardar las dos entidades separadas en dos tablas**, usando el identificador común para cruzar información entre tablas. En este ejemplo, ambas tablas podrían estar unidas por el identificador "Vendedor". En la tabla VENDEDOR, este campo será clave primaria y en la tabla INTERNO, clave secundaria.

Uno-a-Muchos:

- En el caso, VENDEDOR y CLIENTE mantienen relaciones Uno-a-Muchos.
- La relación Uno-a-Muchos **se resuelve agregando la llave primaria de la tabla de Uno (VENDEDOR) a la tabla de Muchos (CLIENTE).**
- Este campo, automáticamente se convierte en **llave secundaria** en la tabla CLIENTE.

**Muchos-a-Muchos:**

- En el caso, CLIENTE y PRODUCTO mantienen relaciones Muchos-a-Muchos.
- Por ejemplo, varios clientes pueden comprar un producto y cada producto puede ser adquirido por varios clientes (el cliente 1 compra detergente y lavandina, mientras que el cliente 2, sólo detergente y el cliente 3, lavandina y escobas)
- Las relaciones Muchos-a-Muchos **no son aceptadas por el sistema.**



En este caso se debe pasar al 4^{TO} PASO: **se deben derivar nuevas tablas.**

¿Qué tablas mantienen relaciones? ¿Qué tipo de relaciones mantienen? ¿Qué tablas mantienen relaciones Muchos-a-Muchos?

3^{ER} PASO: IDENTIFICAR LOS ATRIBUTOS DE LAS ENTIDADES

- En este paso se asignan atributos a las entidades.
- Los atributos son en general **Adjetivos**, características de las entidades que queremos registrar. En el diagrama se muestran como **elipses** atadas a los rectángulos por líneas.
- El sentido común nos dice que los atributos elegidos dependerán del uso que se dará a la base de datos.
- De una misma persona, puede interesarnos grupo y factor sanguíneo, si estamos diseñando el modelo de datos de un Hospital, o la carrera y promedio de notas, si el diseño es para una Universidad.
- En este momento, es fundamental prestar especial atención a los **identificadores**. En una tabla donde no existe identificador, se debe crear uno artificial, que en la mayoría de los casos no será nada más que un número único (Número de... o Código de... son los más comunes).



¿Cuáles son los atributos de cada una de las entidades? ¿Cuál es el identificador?

4^{TO} PASO: DERIVAR NUEVAS TABLAS

- El proceso esencial para organizar datos en tablas relacionales, es llamado **normalización**.
- Consiste en convertir relaciones complejas (Muchos-a-Muchos) en otras más simples que cumplan las condiciones del modelo relacional.

Muchos a Muchos:

- Si la relación es Muchos-a-Muchos, el modelo propone la descomposición de esta relación en dos relaciones Uno-a-Muchos, usando las llaves primarias de las tablas que se quieren relacionar, como atributo en la tabla de conexión.
- Recuerda: ¿CLIENTE y PRODUCTO mantenían relaciones Muchos-a-Muchos?
- En este caso, **se crea una tabla de conexión** que llamaremos COMPRAN. La llave primaria de esta tabla estará compuesta por las **dos llaves primarias de las respectivas tablas**, la de la CLIENTE y la de PRODUCTO.
- La combinación **de ambas llaves constituye la llave principal**.



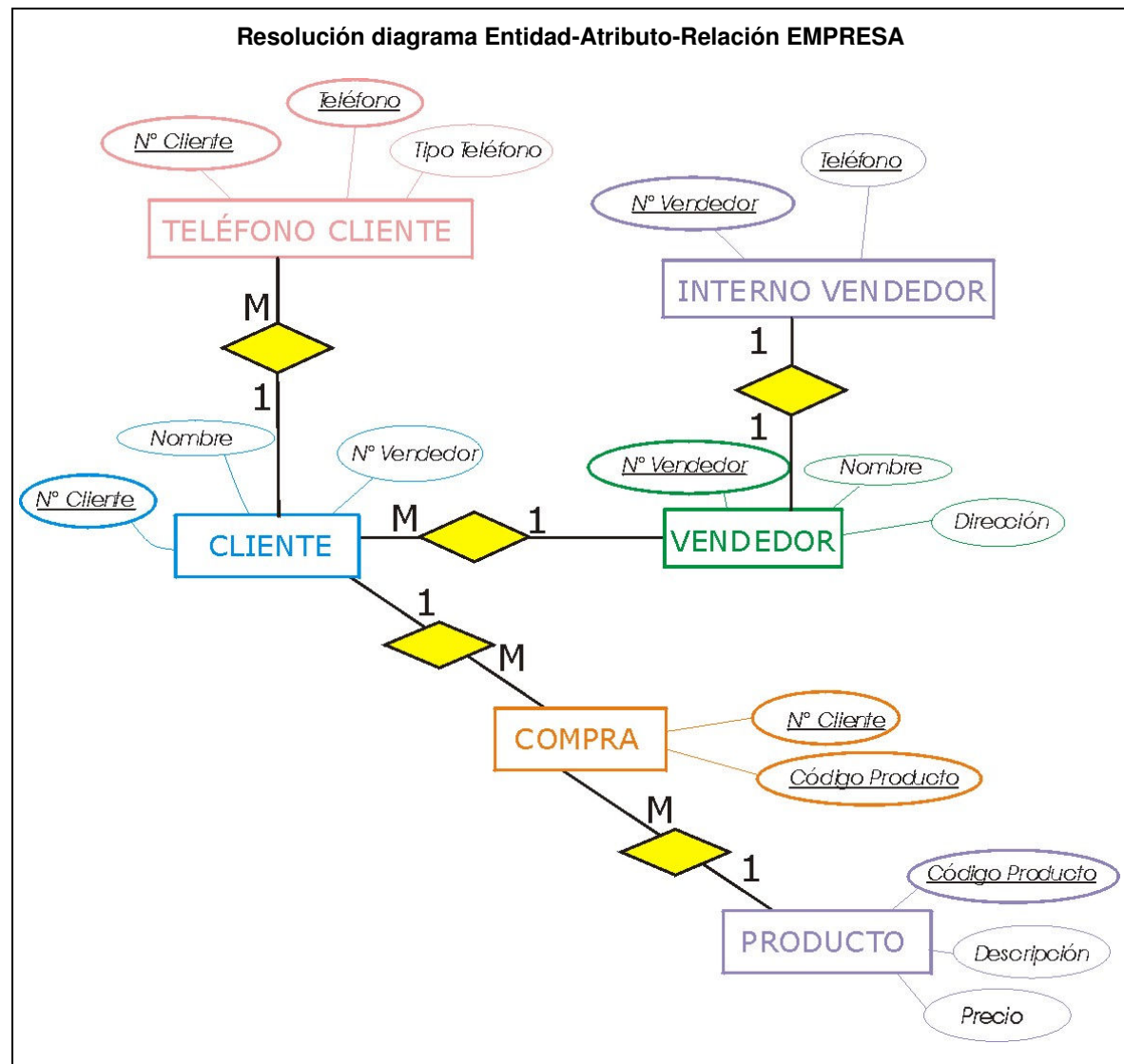
- Esta tabla de conexión intermedia, posibilita convertir una relación Muchos-a-Muchos en relaciones **Uno-a Muchos y Muchos-a-Uno**.

PRODUCTO	PRECIO	CANT	PRODUCTO	CLIENTES	CLIENTES	VENDEDORES
DETERGENTE	3.2	6	DETERGENTE	CLIENTE 1	CLIENTE 1	VENDEDOR 1
ESCOBA	5.9	10	LAVANDINA	CLIENTE 1	CLIENTE 2	VENDEDOR 1
LAVANDINA	1.6	520	DETERGENTE	CLIENTE 2	CLIENTE 3	VENDEDOR 1
		112	ESCOBA	CLIENTE 3		
		11	LAVANDINA	CLIENTE 3		

- Pero ese no es el único caso en que se usa la **normalización**. También se la utiliza para ofrecer una solución al problema denominado **de atributos repetidos**. Por ejemplo, el caso de querer almacenar para cada cliente varios números de teléfono. Tal vez se pregunte ¿Por qué no puedo guardar todos los números en una sola columna, como haría en una ficha papel? Porque el modelo relacional estipula que las entradas deben **ser atómicas**.
- Ante esta restricción, la solución más fácil sería crear tantas columnas como teléfonos pudiera tener un cliente. De esta forma, se crearían las columnas **Teléfono_1**, **Teléfono_2**, **Teléfono_3**, **Teléfono_4** y así sucesivamente.
- Esta solución es totalmente aceptable. ¿Pero que pasaría con un cliente que tuviera 50 teléfonos? ¿Crearía 50 columnas?
- Resumiendo: el problema con esta solución es que crecería en forma desproporcionada la base de datos, ya que habría un montón de columnas que casi no se utilizarían.
- Esta es la solución para este problema que propone la normalización:

TELÉFONO	CLIENTES	CLIENTES	VENDEDORES
4-785-6652	CLIENTE 1	CLIENTE 1	VENDEDOR 1
4-785-5636	CLIENTE 1	CLIENTE 2	VENDEDOR 1
4-544-6963	CLIENTE 2	CLIENTE 3	VENDEDOR 1
4-256-8555	CLIENTE 3		
4-256-9632	CLIENTE 3		

¿Detectó tablas que mantuvieran relaciones Muchos-a-Muchos? ¿Cómo las derivó?
 ¿Se encontró frente al problema denominado de atributos repetidos? ¿Dónde?



Puede consultar el ejercicio de la escuela resuelto <http://www.cursogis.com.ar/BasesP/Bases1.htm>

Las respuestas las encontrará en el link **Resolución Problema**.

5. VENTAJAS Y DESVENTAJAS DEL MODELO RELACIONAL

5.1 Ventajas

- El modelo relacional **es eficiente en la** organización y manejo de grandes colecciones de datos.
- **Permite restricciones de seguridad**, distinguiendo distintos perfiles de usuarios (operador, administrador, Gerente) y asignándole a cada uno de ellos distintos tipos de permisos (sólo lectura, lectura y escritura pero sólo de ciertas tablas, etc.)
- **Posibilitan múltiples vistas de una misma base** según los requerimientos de información del sistema. ¿Recuerda el ejemplo del empleado bancario que no debía acceder a la información de los préstamos?
- **Posibilitan las bases de datos distribuidas**: bases divididas en partes que residen en lugares geográficamente distantes, pero que están fuertemente vinculadas funcionando como una base de datos.
- **Reducen los datos duplicados y redundantes evitando inconsistencias.**
- **Mantienen la integridad y calidad del sistema, ya que los datos están separados de las aplicaciones.** Toda lectura o escritura debe pasar por el DBMS.
- **Permiten el acceso de varios usuarios en el mismo momento.**
- **No requieren conocimientos de programación para acceder a los datos.**

5.2 Desventajas

- **No soportan las relaciones complejas que existen en el mundo real.** Por ejemplo, objetos conformados de otros objetos (ejemplo: objeto “propiedad inmueble” compuesto de los objetos “edificio”, “terreno” y “cerca”).
- **No tienen conocimiento semántico**, es decir, no entienden qué significan los datos que guardan. Por ejemplo, las bases de datos con información hidrológica no “saben” que el agua fluye en declive.
- **Los tipos de datos son limitados** (numéricos, alfanuméricos, etc.) No reconocen tipos de datos más complejos. Para dar un ejemplo trivial, sería bueno que fueran capaces de reconocer un dato de tipo “*Votante*” y que este tipo de dato seleccionará de una base de personas a los individuos mayores de 18 años de edad.
- **Tienen dificultades con el tiempo considerado como una sucesión natural.** Una secuencia de hechos, donde los eventos pasan antes o después de otros, es muy difícil de modelar en una base de datos relacional. En general, sólo se obtiene un registro de la información en determinado momento. La consulta del “*antes*” es siempre difícil.

EVALUACIÓN

- Realice las diferentes ejercitaciones que propone esta clase.
- Si usted es un alumno matriculado:
- Cuando finalice esta tarea, visite <http://www.cursogis.com.ar/Bases/Bases.htm>
- Seleccione el link **Subir trabajo**
- Se abrirá una ventana donde deberá seleccionar la clase 1, llenar los datos respectivos y seleccionar el archivo de evaluación.
- Si no está matriculado y desea mayor información, por favor comuníquese con info@cursogis.com.ar o centrogis@gmail.com
- Podrá consultar los ejercicios resueltos por sus compañeros en <http://www.cursogis.com.ar/BasesP/Bases1.htm>
- Posteriormente, envíe un mail a base@cursogis.com.ar, notificando que su tarea esta finalizada y **solicite la próxima clase.**