

Componente temática.

Introducción

Tal como ya hemos visto en los módulos anteriores, la componente temática permite responder las preguntas: **¿qué?, ¿cuándo?, ¿cómo?**

Son las **características** que se conocen como atributos de los objetos con los que representamos el mundo real. Cada objeto puede registrar un **determinado valor para sus atributos** (variables), los cuales pueden presentar cierta regularidad en el espacio y en el tiempo y, además, pueden ser de distinto tipo y escala de medida.

Estudiaremos los fundamentos básicos de las bases de datos, que son un elemento fundamental en el entorno informático hoy en día y tienen aplicación en la práctica totalidad de campos. Como en módulos anteriores, las estudiaremos según la visión de los Sistemas de Información Geográficas; otros cursos de nivel más especializados permitirán al estudiante acceder a conocimientos más técnicos y específicos.

Concebidas con un propósito general, las bases de datos son de utilidad para toda disciplina o área de aplicación en la que exista una necesidad de gestionar datos, tanto más cuanto más voluminosos sean estos. En los SIG, **los datos son cada día más voluminosos**, debido no solo a una mayor cantidad de información, sino también a una mayor precisión en esta, la cual implica un mayor volumen de datos. Además, presentan otra serie de características (uso múltiple, necesidad de acceso eficiente para análisis, necesidad de indexación, etc.), haciendo todas ellas que sea recomendable el uso de bases de datos y tecnologías específicas para su manejo.

Veremos que los **sistemas de gestión de bases de datos** consisten en una **colección de datos interrelacionados y un conjunto de programas para acceder a los mismos**.

Puede parecer primera vista que esta definición es bastante parecida a la ya hemos visto sobre los Sistemas de Información Geográfica; básicamente

esto se debe a que el núcleo de un SIG se sitúa un sistema de gestión de base de datos.

En primer lugar: Bases de datos

Entendemos como **base de datos** al **conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización.**

Una base de datos puede estar constituida con cualquier tipo de datos: los de tipo puramente espacial (geometrías) al igual que datos numéricos y alfanuméricos como los que constituyen la componente temática de la información geográfica.

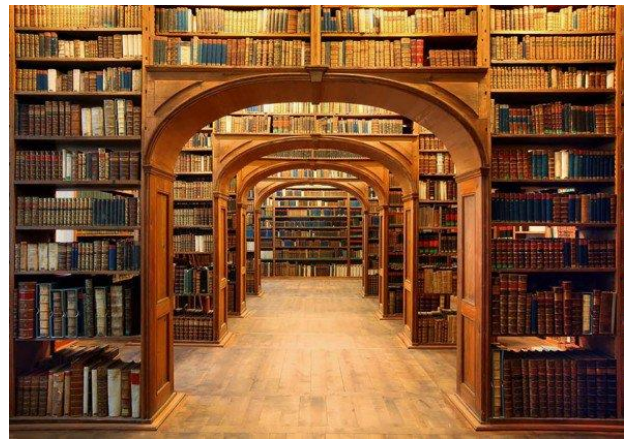
Otras definiciones de base de datos:

- **Conjunto de datos organizado de tal modo que permita obtener con rapidez diversos tipos de información.**
- **Conjunto de datos relacionados entre sí y que tienen un significado implícito.**

Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso; tienen como objetivo el de **satisfacer la necesidad de información de algún ente u organización.**



A modo de ejemplo: una biblioteca puede considerarse una base de datos compuesta en su mayoría por **documentos y textos impresos** en papel e **indexados para su consulta**.



En el ambiente informático, **Crear, Leer, Actualizar** y **Eliminar** (conocidas bajo el nombre de CRUD) son las cuatro funciones básicas de almacenamiento persistente.

A veces se usan palabras alternativas cuando se definen las cuatro funciones básicas de CRUD: **Recuperar** (en lugar de leer), **Modificar** (en lugar de actualizar), **Destruir** (en lugar de eliminar).

Otra definición de base de datos es: **conjunto de información interrelacionada, almacenada digitalmente**.

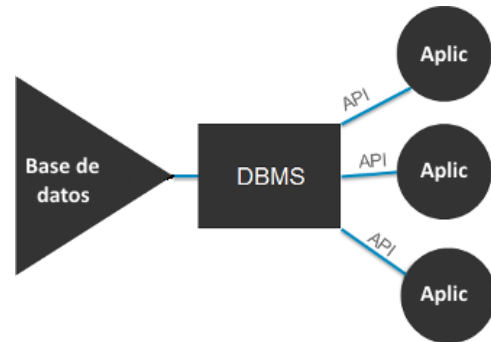


¿Qué es una base de datos? De una manera simple, es un contenedor que permite almacenar la información de forma ordenada con diferentes propósitos y usos.

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. Por ejemplo, combinar la información de las ventas del mes de junio del 2014 para el producto 'X' en la zona norte nos da un indicativo del comportamiento de las ventas en un periodo de tiempo.

Manejadores de Bases de Datos

Un sistema manejador de bases de datos, **Database Management System (DBMS)** es una colección de **software específico**, cuya función es servir de **interfaz** entre la base de datos, el usuario y las distintas aplicaciones utilizadas.



A grandes rasgos, tenemos que el **DBMS** permite a las distintas aplicaciones acceder a la **base de datos**. Para esto utilizan las llamas **APIs** que son clientes que actúan de **interfaz** entre la aplicación y el DBMS.



*En informática, el término **interfaz** se utiliza para nombrar a la conexión funcional entre dos sistemas, programas, dispositivos o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles permitiendo el intercambio de información.*

Como su nombre indica, el objetivo de los sistemas manejadores de base de datos es precisamente el de manejar un conjunto de datos para convertirlos en **información relevante** para la organización, ya sea a nivel operativo o estratégico.

Esto se realiza mediante una serie de rutinas de software que permiten su uso de una manera segura, sencilla y ordenada. Se trata, en suma, de un conjunto de programas que realizan tareas de forma interrelacionada para facilitar la **construcción y manipulación de bases de datos**, adoptando la forma de interfaz entre éstas, las aplicaciones y los mismos usuarios.

Su uso permite realizar un **mejor control** a los administradores de sistemas y, por otro lado, también **obtener mejores resultados a la hora de realizar consultas** que ayuden a la gestión empresarial mediante la generación de la tan perseguida ventaja competitiva.

Un sistema DBMS es sinónimo de **independencia, una redundancia mínima, consistencia** de la información (control de la concurrencia), abstracción de la información sobre su almacenamiento físico, así como un acceso seguro y la adopción de las medidas necesarias para garantizar la integridad de los datos.

Estas particularidades son algunos de los rasgos definitorios de un DMBS, cuyos procesos esenciales son la manipulación y construcción de la bases de datos, así como la definición de los mismos. Son características que, a su vez, facilitan el cumplimiento de una serie de funciones relacionadas con muchos de los aspectos apuntados, entre otros la definición de los datos, su fácil manipulación, una rápida gestión, poder representar relaciones complejas entre datos y otros aspectos relacionados con la seguridad y validez de los datos.

¿Por qué usar un DBMS?

- **Eficiencia:** las bases de datos pueden almacenar gran cantidad de información en forma eficiente, es decir, ocupan menos espacio.
- **Velocidad:** al ser información estructurada, es muy rápido y sencillo encontrar la información.
- Es muy sencillo agregar nuevos datos, editar los existentes y eliminar datos que ya no se utilizan.
- Los datos pueden ser ordenados en forma sencilla.
- **Importación:** Los datos pueden ser fácilmente importados en otras aplicaciones.
- **Acceso múltiple:** más de una persona puede acceder a la misma base de datos al mismo tiempo. Las bases de datos permiten el **control de acceso** a los datos. Los **controles de concurrencia** permiten los mecanismos de bloqueo que facilitan el acceso de múltiples usuarios. No es raro ver bases de datos con miles de usuarios trabajando al mismo tiempo.

- **Replicación:** La replicación está destinada a la copia y distribución de datos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia. La replicación permite distribuir datos entre diferentes ubicaciones y entre usuarios remotos o móviles mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet.

A modo de ejemplo, muchas empresas públicas utilizan la replicación a la hora de editar sus datos. Las bases de datos son replicadas a nivel departamental, y los usuarios acceden a su base de datos local. Una vez por semana, los datos son sincronizados (en ambos sentidos) con las bases de datos principales, dejando los datos en igualdad de condiciones en todo el país.

- **Seguridad:** Es más seguro almacenar los datos en una base de datos que en archivos independientes. La seguridad permiten la identificación de los usuarios a través de la combinación de **usuario y contraseña**, la definición de **roles y permisos** de esos roles sobre los datos.

A modo de ejemplo, una base de datos puede tener roles de lectura y escritura en forma independiente. Un usuario se identifica con su nombre y contraseña, y se le asigna un rol (editor o lector). Este rol asigna (o no) permisos de escritura a un set de datos para que pueda editar o no los mismos.

- **Consistencia:** los datos en una base de datos pueden cumplir "reglas" que evitan la inconsistencia entre los datos.
- **Seguridad y recuperación** ante fallas: existen muchas herramientas de respaldo de la información de las bases de datos. Ante problemas de fallos, los datos pueden ser recuperados con cierta facilidad.
- Soporte de transacciones **ACID**:
 - **Atomicidad:** Si cuando una operación consiste en una serie de pasos, bien todos ellos se ejecutan o bien ninguno, es decir, las transacciones son completas.
 - **Consistencia:** Es la propiedad que asegura que sólo se empieza aquello que se puede acabar.

- **Aislamiento:** Esta propiedad asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error.
- **Durabilidad:** Esta propiedad asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema y que de esta forma los datos sobrevivan de alguna manera.

¿Por qué NO usar un DBMS?

- **Alto costo en inversión** (hardware, software, capacitación técnica).
- El **costo de administración** del DBMS y de las Bases de Datos.
- **No se justifica usar DBMS cuando los datos son pocos** (se pueden gestionar con planillas).
- **Alta complejidad** a la hora de diseñar los modelos de datos.

¿Qué son las tablas?

La componente temática (información tabular) es la base de las entidades geográficas; **esta componente permite la visualización, la consulta y el análisis de los datos disponibles.**

A primera vista, las tablas están constituidas por **filas** y **columnas**, donde todas las filas de un set de datos tienen las mismas columnas. Estas filas reciben en nombre **registros**, y las **columnas** el nombre de campos.

Veamos un ejemplo:

IdTramo	IdRuta	ProgInicia	ProgFinal	Condicion	LargoKMs	IndRegInt	IndEstSup	DivAdm	Firme
1	1	46827	51516	Sin Evaluar	5	30.8	430.0	Artigas	Sin Evaluar
2	3	28085	40092	Regular	12	23.0	242.7	Colonia	Hormigón
3	5	21125	24385	Bueno	3	14.1	302.9	Montevideo	Concreto Asfáltico
4	1	12132	14712	Muy Bueno	3	47.8	274.8	Salto	Tratamiento Bituminoso
5	3	19171	27271	Malo	8	48.8	329.7	Rocha	Imprimación Reforzada
6	5	33578	42373	Malo	9	36.0	450.3	Rivera	Hormigón
7	1	13825	19979	Regular	6	36.8	173.8	Artigas	Concreto Asfáltico
8	3	30506	36245	Sin Evaluar	6	28.7	103.5	Colonia	Tratamiento Bituminoso
9	5	40754	45361	Bueno	5	43.1	406.6	Montevideo	Imprimación Reforzada
10	1	12985	22632	Sin Evaluar	10	26.7	445.8	Salto	Sin Evaluar



La imagen anterior presenta una tabla que presenta y relaciona información sobre **tramos de vialidad**. Esta tabla consta de 10 registros (las filas) y 10 campos (las columnas). Cada campo corresponde a UN atributo determinado de la tabla.

- **IdTramo** *Identificador del tramo.*
- **IdRuta** *Identificador de la ruta a la que pertenece el tramo.*
- **ProgInicia** *Progresiva inicial (en kilómetros) del tramo.*
- **ProgFinal** *Progresiva final (en kilómetros) del tramo.*
- **Condicion** *Condición del tramo.*
- **LargoKMs** *Largo (en kilómetros) del tramo.*
- **IndRegInt** *Índice de regularidad internacional del tramo.*
- **IndEstSup** *Índice de estado superficial del tramo.*
- **DivAdm** *División administrativa (departamento) del tramo.*
- **Firme** *Tipo de firme (material) del tramo.*

Cada registro en forma independiente, corresponde a la información de UN tramo para cada uno de los campos (atributos) definidos en la tabla. Este concepto es muy importante, si no lo comprendemos, difícilmente podamos seguir adelante con este módulo, por lo tanto vamos a profundizar este concepto:

“Las tablas son objetos fundamentales de una base de datos porque en ellas es donde se conserva toda la información o los datos. Así, una base de datos de una empresa puede tener una tabla llamada “Contactos” que almacene los nombres de los proveedores, las direcciones de correo electrónico y los números de teléfono.

*Una base de datos, normalmente tiene **varias tablas relacionadas**. En una base de datos bien diseñada, cada tabla almacena datos sobre un tema en concreto, como empleados o productos. Una tabla tiene **registros** (filas) y **campos** (columnas). Los campos tienen diferentes tipos de datos, como texto, números, fechas e hipervínculos.” (Extraído del sitio de Microsoft).*

Cada campo (columna) puede almacenar UN SOLO TIPO DE DATOS; a modo de ejemplo: un número, una fecha, un texto, un valor lógico.

Actualmente, los sets de datos (es decir, las capas de datos geográficos) son **tablas con campos especiales que contienen información sobre la geometría de las entidades**.

Una de las tareas habituales de los GIS es la posibilidad de asociar registros en una tabla con registros en otra tabla a través de un campo común, conocido como una clave. Esta tarea recibe el nombre de **Unión de Tablas** (o **JOIN**). Estas asociaciones pueden ser llevadas a cabo de varias formas, como unir o relacionar tablas de manera temporal en el mapa o crear uniones permanentes que pueden almacenadas en el disco de una computadora. Este tema lo vamos a ver en detalle en las próximas secciones de este módulo.

Definición de Registro

En el contexto de una base de datos, un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla.

En otras palabras, ya hemos comentado que una tabla de una base de datos puede imaginarse como un conjunto de filas y columnas. **Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura.**

Un registro es un conjunto de campos que contienen los datos que pertenecen a una misma repetición de entidad, en nuestro ejemplo anterior cada registro representa un tramo de vialidad.



IdTramo	IdRuta	ProgInicia	ProgFinal	Condicion	LargoKMs	IndRegInt	IndEstSup	DivAdm	Firme
1	1	46827	51516	Sin Evaluar	5	30.8	430.0	Artigas	Sin Evaluar
2	3	28085	40092	Regular	12	23.0	242.7	Colonia	Hormigón
3	5	21125	24385	Bueno	3	14.1	302.9	Montevideo	Concreto Asfáltico
4	1	12132	14712	Muy Buenc	3	47.8	274.8	Salto	Tratamiento Bituminoso

En la imagen anterior, un registro corresponde a la totalidad de valores que corresponden a cada uno de los campos de la tabla.

A cada registro se le asigna automáticamente un número consecutivo (número de registro) que en ocasiones es usado como índice aunque lo normal y práctico es asignarle a cada registro un campo clave para su búsqueda.

Definición de Campo

Un campo tiene algunas **tipo de datos** características esenciales, como un **nombre** que lo identifica de manera única dentro de una tabla, un que define la naturaleza de los datos, las operaciones que se pueden realizar en los datos y la cantidad de espacio de almacenamiento reservada para cada valor.

IdTramo	IdRuta	ProgInicia	ProgFinal	Condicion	LargoKMs	IndRegInt	IndEstSup	DivAdm	Firme
1	1	46827	51516	Sin Evaluar	5	30.8	430.0	Artigas	Sin Evaluar
2	3	28085	40092	Regular	12	23.0	242.7	Colonia	Hormigón
3	5	21125	24385	Bueno	3	14.1	302.9	Montevideo	Concreto Asfáltico
4	1	12132	14712	Muy Buenc	3	47.8	274.8	Salto	Tratamiento Bituminoso
5	3	19171	27271	Malo	8	48.8	329.7	Rocha	Imprimación Reforzada

En la imagen anterior, el campo señalado corresponde a la información de largo en kilómetros de cada uno de los campos.

Eventualmente un registro puede tener un **valor nulo** (es decir, no existe un valor) en un campo determinado. No debemos confundir valor nulo con valor cero.

“Null (nulo) es un marcador especial que permite indicar que no existe un valor dentro de una base de datos. Introducido por el creador del modelo relacional de bases de datos E. F. Codd, su

función es la de solventar el requisito de que los DBMS puedan representar información **desconocida** o **no aplicable**."

A modo de ejemplo, en una tabla de información parcelaria rural podemos incluir los siguiente campos: **Departamento, Padrón, Superficie, Valor Real, Frentista**.

Departamento	Padron	Superficie	ValorReal	Frentista
Montevideo	13456	3416	3475136	Si
Salto	45621	4146	74262755	No
Canelones	4531	8631	Null	No
Tacuarembó	4597	Null	4521646	Null
Durazno	942	7525	215367	No
Salto	14643	523	7573145	Si
Canelones	7563	Null	Null	No

Es posible, que para algunas parcelas rurales no tengamos la información de alguno de los campos. No es de extrañar que los valores reales no se carguen al mismo tiempo de dar un alta de una parcela en la tabla. O que los datos de superficie no hayan sido determinados aún. En ambos casos se puede utilizar en valor Null tal como se muestra en la imagen anterior.

Otra elemento a tener en cuenta es: **es importante planificar la forma que podemos usar la información almacenada en los campos**.

Por ejemplo, en lugar de almacenar el nombre de una persona en un único campo, es preferible almacenar los componentes en campos independientes para el nombre y apellido. Por lo general, si necesita informar, ordenar, buscar o calcular en un elemento, coloque dicho elemento en un campo por sí mismo.

En lugar de utilizar una tabla con tres campos:

Nombre Completo	Edad	Sexo
Eduardo Santos	54	Masculino
Esther García	13	Femenino
Miguel Olivera	45	Masculino
Carlos Gianpietro	67	Masculino
Gloria Neyra	34	Femenino
Laura Portillo	2	Femenino
Martin Berger	57	Masculino

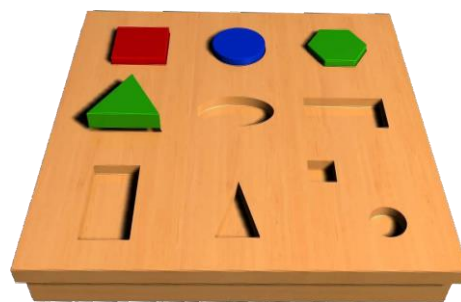
Es preferible una tabla con cuatro campos:

Nombre	Apellido	Edad	Sexo
Eduardo	Santos	54	Masculino
Esther	García	13	Femenino
Miguel	Olivera	45	Masculino
Carlos	Gianpietro	67	Masculino
Gloria	Neyra	34	Femenino
Laura	Portillo	2	Femenino
Martin	Berger	57	Masculino

En conclusión, tener más cantidad de campos permite – generalmente - realizar mayor cantidad de consultas sobre la misma tabla sin necesidad de apelar a funciones específicas.

Tipos de datos de los campos

Al momento de diseñar una tabla o capa geográfica debemos seleccionar un único tipo de datos para cada campo. Los tipos disponibles dependen de cada manejador de base de datos, pero en general incluyen diversos tipos de números, tipos de texto, tipos de fecha, objetos binarios grandes (BLOB) o identificadores únicos a nivel mundial (GUID). **La elección del tipo de datos correcto permite almacenar los datos adecuadamente y facilita las necesidades de análisis, de administración de datos y de negocios.**



De la misma forma que en los juegos de encastrar, al diseñar una tabla, debemos tener en cuenta el **tipo de dato almacenaremos en un campo** y el **tamaño del dato que almacenaremos en el campo**.

Los tipos de datos que veremos en esta sección incluyen algunos de los tipos de datos principales disponibles al crear una capa o tabla con ArcGIS y QGIS. Pero recordemos que esto puede variar dependiendo del DBMS que utilizaremos para almacenar la información

Valores Tipo Número

Como ya hemos comentado en párrafos anteriores, cada campo tiene un tipo de datos relacionado.

Un tipo de datos es un atributo que especifica el tipo de datos que el objeto puede contener: *datos de enteros, datos de números decimales, datos de caracteres (texto), datos de moneda, datos de fecha y hora, cadenas binarias, etc.*

Cada manejador de base de datos proporciona un conjunto de tipos de datos del sistema que define todos los tipos de datos que pueden utilizarse.

Todo dato es de un tipo concreto, cuando diseñamos una tabla debemos indicar los campos para ubicar un tipo de dato concreto. Por otro lado debemos establecer el tamaño máximo para cada campo.

Es posible almacenar números en uno de los cuatro tipos de datos numéricos: **Entero corto**, **Entero largo**, **Flotante** (números de punto flotante de precisión simple), **Doble** (números de punto flotante de precisión doble).

Al elegir el tipo de datos, primero debemos tener en cuenta la necesidad de los **números enteros** versus los **números fraccionarios (o decimales)**.

Si necesitamos almacenar números enteros, como 12 ó 12.345.678; debemos utilizar un entero corto o largo. Si necesita almacenar números fraccionarios que tienen lugares decimales, como 0,23 ó 1234,5678; debemos especificar un flotante o un doble.

Por otro lado, cuando elijamos entre un entero corto o largo, o entre flotante o doble, debemos elegir el tipo de datos que ocupe el menor espacio de almacenamiento posible.

Esto no sólo permite minimizar la cantidad de espacio de almacenamiento necesario sino que también mejorará el rendimiento. **Esto se relaciona directamente con el modelo de almacenamiento que presentamos en módulos anteriores.**

Si necesitamos almacenar números enteros entre -32.768 y 32.767, podemos especificar el tipo de datos entero corto, ya que solo requiere 2 bytes, mientras que el tipo de datos entero largo requiere 4.

Si necesitamos almacenar números fraccionarios entre -3,4E38 y 1,2E38, podemos especificar el tipo de datos flotante, ya que solo requiere 4 bytes, mientras que el tipo de datos doble requiere hasta 8.



Tipo de datos	Rango de almacenamiento	Tamaño (Bytes)	Aplicaciones
Entero corto	-32.768 a 32.767	2	Valores numéricos sin valores fraccionales dentro de un rango específico; valores codificados
Entero largo	-2.147.483.648 a 2.147.483.647	4	Valores numéricos sin valores fraccionales dentro de un rango específico
Flotante (número de punto flotante de precisión simple)	aproximadamente -3,4E38 a 1,2E38	4	Valores numéricos con valores fraccionales dentro de un rango específico
Doble (número de punto flotante de precisión doble)	aproximadamente -2,2E308 a 1,8E308	8	Valores numéricos con valores fraccionales dentro de un rango específico

Dependiendo del software y del DBMS, podemos llegar a tener la necesidad de especificar la **precisión** (la longitud máxima del campo) y la **escala** (el número máximo de posiciones decimales).

La especificación de la precisión y la escala nos permite restringir el rango de valores y los formatos de números que un campo puede aceptar, lo que brinda mayor control.

A modo de ejemplo, si especificamos un número de punto flotante con una precisión de 4 y una escala de 2, el campo aceptará 12,34. Si intentamos introducir 12,345 en el campo, se mostrará un mensaje de error ya que esto supera el número máximo de dígitos y posiciones decimales permitidos. Por otro lado, si especificamos un número de punto flotante con una precisión de 5 y una escala de 3, el campo le permitirá introducir 12,345.



Texto

Un campo de tipo texto representa una **serie de símbolos alfanuméricos**. Esto puede incluir nombres de calles, propiedades de atributos u otras descripciones textuales.

Una alternativa para repetir atributos textuales en una base de datos es establecer un valor codificado. Una descripción textual se codificará con un valor numérico.

Por ejemplo, podemos codificar tipos de carreteras con valores numéricos asignándole un 1 a las carreteras de pavimento mejoradas, un 2 a las carreteras de ripio, y así sucesivamente.

Esto tiene la ventaja de que utilizaremos menos espacio de almacenamiento en la base de datos; sin embargo, los usuarios deben conocer la manera de interpretar los valores codificados.

Los caracteres utilizados para el texto varían según el idioma.

Fechas

En el tipo de datos de fecha podremos almacenar fechas, horas o fechas y horas. El formato predeterminado en el cual se presenta la información es **mm/dd/aaaa hh:mm:ss** y una especificación para a.m. o p.m.

BLOB

Un BLOB son datos almacenados como una secuencia larga de números binarios. En ArcGIS y QGIS, las anotaciones y dimensiones, las imágenes, multimedia o bits de código, se pueden almacenar en este tipo de campo. Es necesario que contar con un cargador o un visualizador personalizado, o una aplicación de terceros, para cargar elementos en un campo BLOB o para visualizar el contenido de un campo BLOB.

A modo de ejercicio

Indicar, a partir de la siguiente tabla, el nombre, tipo, precisión y escala de cada uno de los campos que identifique.

IdTramo	ProgInicia	ProgFinal	LargoKMs	IndRegInt	IndEstSup	DivAdm	IndServ	CatTran	Zona1	EstCons	LongIni	LatIni	LongFin	LatFin
1	46827	51516	5	30.8	430.0	Artigas	16.6	Categoría 3	ZonaA	2	-53.738346	-29.629071	-51.068030	-34.233024
2	28085	40092	12	23.0	242.7	Colonia	22.8	Categoría 2	ZonaB	2	-54.083118	-31.200099	-51.870097	-29.744769
3	21125	24385	3	14.1	302.9	Montevideo	10.1	Categoría 1	ZonaC	5	-54.906973	-34.408958	-49.526799	-29.079185
4	12132	14712	3	47.8	274.8	Salto	24.0	Categoría 3	ZonaD	1	-50.836109	-34.054939	-50.870187	-34.558944
5	19171	27271	8	48.8	329.7	Rocha	38.1	Categoría 1	ZonaA	2	-54.320443	-32.501308	-51.527379	-33.094549
6	33578	42373	9	36.0	450.3	Rivera	26.9	Categoría 5	ZonaB	3	-49.750404	-30.355168	-52.591046	-31.583840
7	13825	19979	6	36.8	173.8	Artigas	25.6	Categoría 2	ZonaC	2	-49.017523	-34.162867	-52.339236	-33.307200
8	30506	36245	6	28.7	103.5	Colonia	20.6	Categoría 5	ZonaD	5	-51.270688	-32.084158	-53.866554	-32.537673
9	40754	45361	5	43.1	406.6	Montevideo	15.0	Categoría 2	ZonaA	4	-51.073071	-30.647429	-50.279128	-31.192447
10	12985	22632	10	26.7	445.8	Salto	33.9	Categoría 3	ZonaB	4	-54.678729	-33.795619	-53.368252	-33.138089

Definición de Campos Claves

En las base de datos los registros de las diferentes tablas deben estar perfectamente identificados; para esto utilizamos los **campos claves**.

Haciendo un paralelismo con la vida real, cada persona tiene una cédula de identidad, pueden existir dos personas con igual nombre e incluso apellidos iguales, pero ambos se diferenciarán por su cedula de identidad. Es decir, **la relación entre una persona y su cédula de identidad es de 1 a 1**; una persona tiene una única cédula de identidad y una cédula corresponde únicamente a una persona. Podemos decir que al hablar de una cédula de identidad hacemos mención a una persona, y viceversa.

Por lo general, **es importante que cada tabla contenga al menos un campo que permita identificar unívocamente cada registro**. Este campo puede ser real, como la propia cédula de identidad de una persona. O puede ser un valor que nos inventemos para conseguir el mismo fin, como un número que asignemos automáticamente a cada registro, asegurándonos de que no se repita.

Otra opción es utilizar una combinación de campos para identificar los registros de la tabla. Por ejemplo podríamos definir la combinación de

Nombre y Apellidos de nuestra tabla para este fin. Sin embargo, existe la probabilidad de que existan alumnos con igual nombre y apellidos, por lo que no sería una clave del todo válida.



*En la tabla del ejercicio anterior, ¿cuál podría ser un campo clave?
¿Por qué el campo ESTCONS no puede ser un campo clave?*

¿Qué son las “relaciones”?

Si bien existen varios tipos de base de datos, las más utilizadas actualmente son las llamadas “relacionales”, que han demostrado su amplia utilidad en la mayor parte de situaciones.

Estas bases de datos relacionales, las encontraremos en nuestros sistemas de información geográfica; por este motivo, haremos una breve introducción a este tema.

El modelo relacional fue desarrollado en 1969, y consiste básicamente en un conjunto de **relaciones tabulares**. Las relaciones son tan importantes como los propios datos (las tablas, en este caso), y constituyen una idea central en el modelo relacional, de ahí su denominación.



Durante mucho tiempo lo más habitual era utilizar el SGBD para almacenar la información temática y el SIG para la información geométrica y topológica. Una de las funcionalidades de este modelo era el enlazado de ambos tipos de información que se almacenaban de formas completamente diferentes. Se trata del modelo de datos geo-relacional.

Actualmente, si bien el nombre se conserva, podemos almacenar ambas componentes (espacial y temática) en el mismo SGBD.

Las denominaciones de tabla, fila y columna, cambian a una terminología específica empleada al referirse a las bases de datos relacionales. Es así que la tabla se conoce como **relación**, el registro (o fila) se denomina **tupla**, el campo (o columna) se denomina **atributo**, la cantidad de registros se denomina **cardinalidad**, los valores posibles que puede tener un campo se denomina **dominio**.

Terminología habitual	Modelo relacional
Tabla	Relación
Fila	Tupla
Columna	Atributo
Número de filas	Cardinalidad
Valores posibles	Dominio
Identificador único	Clave primaria

Esquema de la relación

Una forma abreviada de definir las relaciones que forman parte de una base de datos es **mediante su nombre y su esquema** expresado como una lista de los atributos que lo constituyen.

Por ejemplo: podemos definir una relación denominada PERSONAS como:

PERSONAS (CI, Nombre, Altura, Edad, Ciudad)

Según el autor Victor Olaya, *"el esquema de la relación está formado por los nombres de los atributos y un dominio asociado a estos, que delimita el rango de valores posibles para cada atributo. El dominio especifica el tipo de dato a contener en cada columna."*



El dominio está directamente asociado al tipo de dato del atributo.

En una relación, **el orden de tuplas y el orden de los atributos son irrelevantes** (exceptuando la cabecera, que no es un tupla como tal, sino que define el esquema como hemos visto), pero es importante que cada atributo sea del tipo correspondiente a la columna a la que pertenece. Esto lo conoceremos como la **coherencia del esquema**.

Una forma abreviada de definir las relaciones que forman parte de una base de datos es mediante su nombre y su esquema expresado como una lista de los atributos que lo constituyen.

Como ya hemos comentado, una base de datos contiene normalmente más de una tabla, ya que suelen ser muchos los tipos de datos a almacenar y resulta conveniente dividirlos en distintas tablas. Además de las relaciones

que la tabla en sí implica, es necesario definir relaciones entre las distintas tablas, y para ello se emplean los denominados atributos *clave*.

Como ya comentamos en otra sección, un atributo clave es aquel que tiene valor único para cada tupla, pudiendo servir para representar a esta plenamente.

Por convención, los atributos claves se escriben subrayadas al definir el esquema de la tabla, de tal modo que la relación PERSONAS quedaría de la siguiente forma:

PERSONAS (CI, Nombre, Altura, Edad, Ciudad)

Dejaremos este tema por acá. Como ya hemos comentado al principio de este curso, los equipos interdisciplinarios son una de las bases de los sistemas de información geográfica. El modelo relacional es muy utilizado en el ambiente informático, lugar donde seguramente llevaremos adelante esta disciplina. Por este motivo debemos conocer y saber otras formas de denominación.

Fuentes de información tabular

Existen muchas fuentes de datos tabulares, **depende de cada uno de los softwares GIS que puedan aprovechar o no estos formatos**. Obviamente, los GIS propietarios pueden acceder a un conjunto muy amplio de formatos tabulares, mientras que los GIS libres tienen menos formatos. Más allá de esto, los formatos básicos (*que son los que veremos en este curso*) pueden ser accedidos desde cualquiera de las plataformas.

Esta información tabular puede almacenarse en una variedad de recursos (siempre) informáticos: **tablas en carpetas, bases de datos, archivos de texto, consultas en bases de datos**, etc. Iremos viendo paulatinamente alguno de estos conceptos.

Las tablas en carpeta pueden ser tablas de **dBASE**, archivos **INFO**, archivos de **texto**, archivos de **Microsoft Excel**,

Las tablas de la base de datos pueden incluir algunos tipos de información que no son compatibles con las tablas basadas en archivos. A modo de

ejemplo, las tablas de la base de datos pueden almacenar tipos de campos BLOB o ráster. Además, las bases de datos ofrecen capacidades para ampliar la funcionalidad de las tablas, como el mantenimiento de la integridad de datos y la administración de las transacciones.

Tablas en formato dBASE

Los archivos de dBASE (archivos DBF) son un claro ejemplo de lo que denominamos la **“Primera generación de BASE DE DATOS en los SIG”**, donde la información se almacenaba en archivos independientes.

Los primeros programas SIG se caracterizaban por una ausencia completa de cualquier tipo de almacenamiento estructurado. En estas aplicaciones, los datos no se veían como un elemento más dentro de un sistema, sino como una parte del propio software (al menos como algo asociado únicamente a un producto particular). **La práctica habitual era uso de archivos con formatos cerrados, pensados para ser leídos y escritos casi de forma exclusiva por la aplicación particular que ha de consumirlos.** En este sentido los archivos DBF fueron diseñados, como comentaremos más adelante en este módulo, para ser utilizados dentro del software dBASE. Lo mismo ocurre con los archivos Shapefiles; estos fueron creados para extender los DBF para permitir almacenar información geográfica; en su momento fueron creados para ser utilizados por un software específico: Arcview. Ejemplo similar de esta primera generación son los archivos DWG; fueron creados para ser utilizados solamente en Autocad.

En esta primera generación de bases de datos, **integrar en el SIG otros datos distintos a aquellos para los que la aplicación no se había diseñado no era sencillo.** Existía una vinculación muy directa entre software y datos. Asimismo, las funcionalidades del software eran también específicas para esos datos, y todas ellas se implementaban directamente en la aplicación.

Al no existir un DBMS que se encargara de gestionar las operaciones, era el propio SIG quien debía ser responsable de las funcionalidades de acceso o edición.

Otras funcionalidades típicas de un DBMS, sin embargo, no aparecían en estos primeros SIG, ya que no eran necesarias. Por ejemplo: la integridad de los datos en operaciones concurrentes de varios usuarios no era necesario.

Una de las ventajas que pueden encontrarse en este enfoque son las relacionadas con el rendimiento, que podía en ciertos casos ser mayor que el esperable en caso de utilizar un DBMS para canalizar el trabajo con los datos. Son todavía útiles para proyectos de pequeña escala, o durante la etapa de aprendizaje.

Ahora bien, volviendo a los archivos **DBF**, el software **dBASE** fue uno de los primeros sistema de gestión de base de datos usado ampliamente para microcomputadoras.

La **versión III Plus** se convirtió en uno de los títulos de software más vendidos durante un buen número de años. Originalmente se creó bajo una interfaz de texto (las viejas ventanas de fondo negro y letras color verde).



```
Press the F1 key for HELP.
. use f:exemplo
. list stru
Structure for database: f:exemplo.dbf
Number of data records: 4
Date of last update : 06/01/93
Field  Field Name  Type      Width  Dec
1  CODIGO      Numeric   6
2  NOME        Character 30
3  ENDERECO    Character 30
4  BAIRRO      Character 20
5  CIDADE      Character 20
6  ESTADO      Character  2
7  CREDITO     Numeric   9      2
8  SITUACAO    Logical   1
** Total **                119

Command Line  |<S>| EXEMPLO | Rec: 1/4 | Num
Type a command (or ASSIST) and press the ENTER key (↵).
Enter a dBASE III PLUS command.
```

Posteriormente, evolucionó a la interfaz gráfica establecida por **Microsoft Windows** terminó siendo desplazado por otros productos como Paradox, Clipper, y FoxPro.

El software dBase se creó antes de que **SQL (Structured Query Language**, lenguaje específico que da acceso a un sistema de gestión de bases de datos relacionales) fuera popular y un estándar, en su lugar incorporaba un lenguaje propio y era capaz de permitir a múltiples personas compartir y editar una misma base de datos en una red local.

En 1988 llegó la **versión IV**; todas las ediciones se licenciaban a los usuarios por un plazo de quince años (se entendía como "inconcebible" el hecho de que un usuario utilizara su copia de dBASE por tan largo período).

El software dBASE fue vendido a la empresa **Borland** a comienzos de la década del 90. Al poco tiempo promovió una casi intrascendente **versión 5**, de la que llegó a haber versión para Windows.

El formato subyacente de dBASE, el archivo dbf, es ampliamente utilizado en muchas otras aplicaciones que necesitan un formato simple para almacenar datos estructurados, esto es la base de los software SIG.

¿Por qué son importantes en un SIG? Como ya comentamos, los archivos Shapefiles (que ya hemos visto en clase) se basan en los archivos DBF.

Durante la década del 80 los archivos DBF eran prácticamente un standard; por este motivo ESRI extiende este formato incorporando la posibilidad de almacenar (en un archivo externo) las geometrías de los elementos geográficos. Ya que estamos en este tema veremos en esta sección como está compuesto un archivo Shapefile.

Según la empresa ESRI (creador de este formato de archivo), *“un shapefile es un formato sencillo y no topológico que se utiliza para almacenar la ubicación geométrica y la información de atributos de las entidades geográficas. Las entidades geográficas de un shapefile se pueden representar por medio de puntos, líneas o polígonos (áreas). El espacio de trabajo que contiene shapefiles también puede incluir tablas del dBASE, que pueden almacenar atributos adicionales que se pueden vincular a las entidades de un shapefile.”*

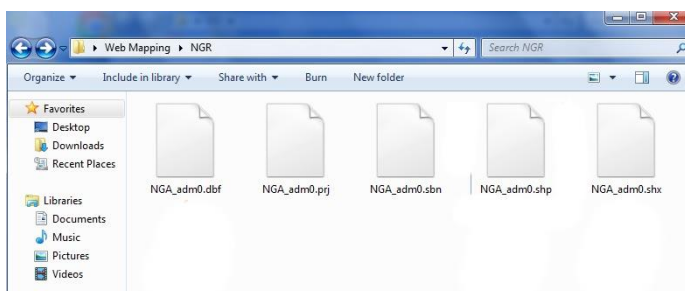
Los shapefiles poseen un formato sencillo y no topológico, y sirven para almacenar la ubicación geométrica y la información de atributos de las entidades geográficas. Un shapefile es uno de los formatos de datos espaciales con las que puede trabajar y editar en ArcGIS.

El formato SHP define la geometría y los atributos de entidades a la que se hace referencia geográfica en tres o más archivos con extensiones de archivo concretas que se almacenan en la misma carpeta de trabajo. Éstas son:

- .shp** es el archivo principal que almacena la geometría de la entidad; necesario.
- .shx** es el archivo de índice que almacena el índice de la geometría de la entidad; necesario.

- .dbf** es la tabla dBASE que almacena la información de atributos de las entidades; necesaria.
Entre la geometría y los atributos existe una relación de uno a uno, basada en el número de registro. Los registros de atributo del archivo dBase deben estar en el mismo orden que registros del archivo principal.
- .sbn y .sbx** son los archivos que almacenan el índice espacial de las entidades.
- .fbn y .fbx** son los archivos que almacenan el índice espacial de las entidades para los shapefiles que son solo de lectura.
- .ain y .aih** son los archivos que almacenan el índice de atributo de los campos activos en una tabla o una tabla de atributos del tema.
- .ixs** índice de geocodificación para los shapefiles de lectura y escritura.
- .mxs** índice de geocodificación para los shapefiles de lectura y escritura (formato ODB).
- .prj** es el archivo que almacena información del sistema de coordenadas.
- .xml** metadatos de ArcGIS, es el archivo que almacena información sobre el shapefile.
- .cpg** es un archivo opcional que se puede utilizar para especificar la página de código para identificar el conjunto de caracteres que se va a utilizar.

Cada archivo debe tener el mismo prefijo, por ejemplo Parcelas.shp, Parcelas.shx, Parcelas.dbf, Parcelas.prj, etc. Pueden estar (o no) presentes todos los archivos.



Los archivos MINIMOS para que un shapefile tenga sentidos son: shp, shx, dbf y prj. Es decir, la componente geográfica (shp), la componente temática (dbf), la relación entre ambos (shx) y el sistema de referencia.

Tablas en Archivos de Texto

Los archivos de texto **CSV (archivos separados por comas)** son un tipo de documento en formato abierto y sencillo que permite almacenar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea. Es decir, **cada renglón es una línea y cada campo está separado por comas.**

```
Año,Marca,Modelo,Descripción,Precio
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture Extended Edition","",4900.00
1999,Chevy,"Venture Extended Edition, Very Large",,5000.00
1996,Jeep,Grand Cherokee,"MUST SELL! air, moon roof, loaded",4799.00
```

El formato CSV **no está estandarizado**: la idea básica de separar los campos con una coma es muy clara y sencilla, pero se vuelve complicada cuando el valor del campo también contienen comillas dobles o saltos de línea.

Los distintos tipos de archivos CSV pueden no manejar esos datos, o usar comillas otra clase para envolver el campo. Pero esto no resuelve el problema: algunos campos también necesitan embeber estas comillas, así que los distintos tipos de CSV pueden incluir caracteres o secuencias de escape.

Actualmente, el término CSV también denota **otros formatos de valores separados por delimitadores que usan delimitadores diferentes a la coma** (como los valores separados por tabuladores). Un delimitador que no está presente en los valores de los campos (como un tabulador) mantiene el formato simple.

Estos archivos separados por delimitadores alternativos reciben en algunas ocasiones la extensión CSV aunque este uso sea incorrecto. Esto muchas veces causa problemas en el intercambio de datos, por ello muchas aplicaciones que usan archivos CSV tienen opciones para cambiar el carácter delimitador.

A modo de ejemplo, la UNASEV presenta información de uso público a través de archivos CSV. El siguiente [link](#) permite descargar información de las **Personas fallecidas en siniestros de tránsito**.

```
FECHA Y HORA;DEPARTAMENTO;LOCALIDAD;JURISDICCION;TIPO DE SINIESTRO;VEHICULO;ROL;EDAD;SEXO;FALLECIDO A LOS (Días);OTRO VEHICULO;LUGAR DEL SINIESTRO;X;Y
01/01/2014 07:30;CERRO LARGO;ISIDORO NOBLIA;DEPARTAMENTAL;DESPISTE;AUTO;CONDUCTOR;26;M;1;1;CAMINO SAN DIEGO KM 63;771431,5469;6459626,702
01/01/2014 22:58;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;41;M;2;CAMION;CAMINO DOMINGO ARENA esq. AVENIDA DON PEDRO DE MENDOZA;577116,5057;6146717,417
02/01/2014 17:20;TREINTA Y TRES;N/C;NACIONAL;DESPISTE;AUTO;CONDUCTOR;31;M;0;1;RUTA 17 km. 303;758521,3639;6325115,787
03/01/2014 08:50;CANELONES;CANELONES;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;BICICLETA;CONDUCTOR;9;M;0;CAMION;DOCTOR CRISTOBAL CENDAN esq. CAMINO MELGAREJO;565476,1736;6180421,434
04/01/2014 09:05;CANELONES;N/C;NACIONAL;CAIDA;MOTO;CONDUCTOR;65;M;0;1;RUTA 33 km. 23,200;581103,5889;6156323,483
05/01/2014 11:00;TACUAREMBO;N/C;NACIONAL;COLISION ENTRE VEHICULOS;AUTO;CONDUCTOR;26;M;0;1;RUTA 43 km. 44;606804,2937;6403090,437
05/01/2014 11:00;TACUAREMBO;N/C;NACIONAL;COLISION ENTRE VEHICULOS;AUTO;PASAJERO;21;F;0;1;RUTA 43 km. 44;606804,2937;6403090,437
05/01/2014 18:40;SAN JOSE;N/C;NACIONAL;COLISION CON OBSTACULO;MOTO;PASAJERO;44;M;1;1;RUTA 23 km. 114;490894,8088;6240276,083
06/01/2014 22:30;MALDONADO;PIRIAPOLIS;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;50;M;1;1;CAMIONETA;RUTA 37 esq. MAÚA;658262,6789;6142943,069
07/01/2014 11:40;ARTIGAS;ARTIGAS;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;65;M;2;MOTO;GENERAL JUAN ANTONIO LAVALLEJA esq. SARANDI;550936,3074;6636852,885
09/01/2014 06:45;MALDONADO;MALDONADO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;69;M;2;AUTO;ALBERTO CL CARACARA esq. MANUEL FRANCA;688767,206;6136410,283
09/01/2014 09:00;TREINTA Y TRES;N/C;NACIONAL;DESPISTE;AUTO;CONDUCTOR;58;M;0;1;RUTA 8 km. 315;747706,7007;6346473,916
09/01/2014 09:00;TREINTA Y TRES;N/C;NACIONAL;DESPISTE;AUTO;PASAJERO;53;F;0;1;RUTA 8 km. 315;747706,7007;6346473,916
10/01/2014 17:25;SAN JOSE;N/C;NACIONAL;DESPISTE;CAMION;CONDUCTOR;46;M;0;1;RUTA 3 km. 128,300;527531,9264;6234594,062
11/01/2014 07:00;PAYSANDU;PAYSANDU;DEPARTAMENTAL;CAIDA;MOTO;CONDUCTOR;59;M;3;3;DAPASO ANTONIO LARRAÑAGA esq. INDEPENDENCIA;397691,7628;6424687,185
11/01/2014 08:00;ROCHA;BARRA DE CHUY;DEPARTAMENTAL;DESPISTE;CAMIONETA;CONDUCTOR;23;M;0;1;CAMINO ACCESO A BARRA DE CHUY;832274,7028;6257419,305
11/01/2014 08:00;ROCHA;BARRA DE CHUY;DEPARTAMENTAL;DESPISTE;CAMIONETA;PASAJERO;23;M;0;1;CAMINO ACCESO A BARRA DE CHUY;832274,7028;6257419,305
11/01/2014 08:00;ROCHA;BARRA DE CHUY;DEPARTAMENTAL;DESPISTE;CAMIONETA;PASAJERO;24;M;0;1;CAMINO ACCESO A BARRA DE CHUY;832274,7028;6257419,305
11/01/2014 08:00;ROCHA;BARRA DE CHUY;DEPARTAMENTAL;DESPISTE;CAMIONETA;PASAJERO;25;M;0;1;CAMINO ACCESO A BARRA DE CHUY;832274,7028;6257419,305
11/01/2014 16:20;CANELONES;PANDO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;31;M;23;OMNIBUS;AVENIDA JOSE PEDRO VARELA esq. CESAR PLOVENE;595308,8281;6157714,676
11/01/2014 19:00;CANELONES;TOLEDO;NACIONAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;50;M;0;1;RUTA 33 km. 23;581053,1245;6155940,009
11/01/2014 19:00;CANELONES;TOLEDO;NACIONAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;44;M;0;1;RUTA 33 km. 23;581053,1245;6155940,009
12/01/2014 14:40;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;CAIDA;MOTO;CONDUCTOR;27;M;0;1;RAMBLA EUSKALERRIA esq. CANDELARIA;581131,4951;6139518,344
12/01/2014 18:15;COLONIA;N/C;NACIONAL;COLISION ENTRE VEHICULOS;AUTO;PASAJERO;9;F;1;1;CAMIONETA;RUTA 1 km. 123;474070,8842;6200769,376
12/01/2014 18:15;COLONIA;N/C;NACIONAL;COLISION ENTRE VEHICULOS;AUTO;PASAJERO;1;F;1;1;CAMIONETA;RUTA 1 km. 123;474070,8842;6200769,376
13/01/2014 23:40;COLONIA;BRITOPOLIS;DEPARTAMENTAL;CAIDA;MOTO;CONDUCTOR;18;M;0;1;CAMINO RIMON DEL REY entre BRITOPOLIS Y BLANCARENA;474217,3305;6189703,002
14/01/2014 10:35;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;AUTO;CONDUCTOR;30;M;9;CAMION;CAMINO CARRASCO esq. PIRAN;580191,5788;6140730,498
15/01/2014 16:00;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;33;M;1;1;CAMION;CAMINO GENERAL MAXIMO SANTOS esq. BELL;572301,3584;6145527,449
16/01/2014 15:00;MONTEVIDEO;MONTEVIDEO;NACIONAL;COLISION ENTRE VEHICULOS;MOTO;PASAJERO;38;F;1;1;RUTA 5 esq. CAMINO LA REDENCION;567251,3036;6148838,208
16/01/2014 17:45;CANELONES;N/C;NACIONAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;85;M;0;1;RUTA 80 km. 82;627318,6431;6174663,157
16/01/2014 22:40;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;AUTO;PASAJERO;43;M;1;1;OMNIBUS;HUM esq. GENERAL MARTIN RODRIGUEZ;575467,1729;6143489,037
16/01/2014 22:40;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;AUTO;PASAJERO;40;F;5;1;OMNIBUS;HUM esq. GENERAL MARTIN RODRIGUEZ;575467,1729;6143489,037
18/01/2014 08:27;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;ATROPELLO DE PEATON;PEATON;PEATON;28;F;1;1;OMNIBUS;CAMINO REPETTO esq. CAMINO PASO DE LA ESPAÑOLA;578967,5649;6146962,909
18/01/2014 13:00;RIO NEGRO;N/C;NACIONAL;COLISION ENTRE VEHICULOS;CAMIONETA;CONDUCTOR;42;M;0;1;OMNIBUS;RUTA 3 km. 324;436921,0783;6384814,428
18/01/2014 16:00;FLORIDA;N/C;NACIONAL;DESPISTE;AUTO;PASAJERO;63;F;0;1;RUTA 42 km. 14,200;575502,796;6270848,562
18/01/2014 19:00;COLONIA;ARTILLEROS;DEPARTAMENTAL;DESPISTE;AUTO;CONDUCTOR;37;M;0;1;CALLE 13 esq. CALLE F;446892,7166;6190601,502
19/01/2014 01:15;CANELONES;N/C;NACIONAL;ATROPELLO DE ANIMALES;MOTO;CONDUCTOR;29;M;0;1;RUTA 81 km. 35;589640,4734;6190277,154
19/01/2014 01:15;CANELONES;N/C;NACIONAL;ATROPELLO DE ANIMALES;MOTO;PASAJERO;21;F;0;1;RUTA 81 km. 35;589640,4734;6190277,154
19/01/2014 01:40;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;19;M;0;1;MOTO;MATILDE PACHECO DE BATLLE Y ORDOÑEZ esq. AZOTEA DE LIMA;577941,541;6146238,188
19/01/2014 01:40;MONTEVIDEO;MONTEVIDEO;DEPARTAMENTAL;COLISION ENTRE VEHICULOS;MOTO;CONDUCTOR;19;M;0;1;MOTO;MATILDE PACHECO DE BATLLE Y ORDOÑEZ esq. AZOTEA DE LIMA;577941,541;6146238,188
```



En la imagen anterior, ¿qué campos puede identificar? ¿De qué tipo de datos son?

¿Por qué son importantes en un SIG? Fundamentalmente porque es un formato **MUY SENCILLO y gratuito** (cualquier editor de texto nos permite generar un CSV). En contrapartida, no es posible explicitar los tipos de los campos y los caracteres de escape nos dan dolores de cabeza. Por otro lado, pueden contener información espacial; si observamos con detalle el archivo CSV anterior, vemos que tenemos campos X e Y. Estos contienen la coordenadas X e Y planas de los siniestros de tránsito. Por lo tanto, pueden ser representados geográficamente. El inconveniente es que a simple vista no podemos conocer el sistema de referencia; el mismo debe ser indicado por quien provee los datos. Esto forma parte de los metadatos de la información geográfica.

A modo de ejercicio...

Descargue el archivo CSV de la UNASEV indicado anteriormente; agréguelo como tabla al software de su preferencia; agréguelo como tabla de eventos a partir de los campos X e Y. El sistema de referencia es 32721.

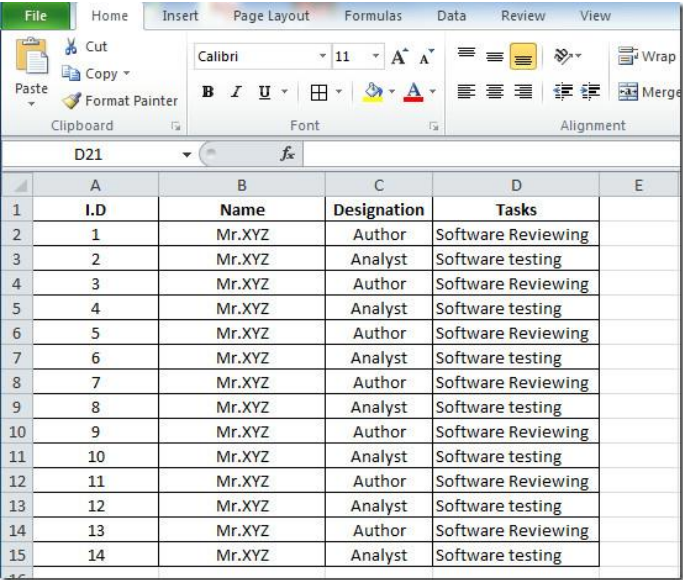
Tablas en archivos de Microsoft Excel

Microsoft Excel es una aplicación de hojas de cálculo que forma parte de la suite de oficina **Microsoft Office**.

Es una aplicación utilizada en tareas financieras y contables, con fórmulas, gráficos y un lenguaje de programación.

Las características, especificaciones y límites de Excel han ido variando considerablemente de versión en versión, exhibiendo cambios en su interfaz operativa y capacidades.

Actualmente, una planilla de Microsoft Excel puede tener 16384 columnas y de 1.048.576 filas por hoja. Esto se traduce en tablas de tamaño medio.



The screenshot shows the Microsoft Excel interface with a table containing 15 rows of data. The table has four columns: I.D, Name, Designation, and Tasks. The data is as follows:

	A	B	C	D	E
1	I.D	Name	Designation	Tasks	
2	1	Mr.XYZ	Author	Software Reviewing	
3	2	Mr.XYZ	Analyst	Software testing	
4	3	Mr.XYZ	Author	Software Reviewing	
5	4	Mr.XYZ	Analyst	Software testing	
6	5	Mr.XYZ	Author	Software Reviewing	
7	6	Mr.XYZ	Analyst	Software testing	
8	7	Mr.XYZ	Author	Software Reviewing	
9	8	Mr.XYZ	Analyst	Software testing	
10	9	Mr.XYZ	Author	Software Reviewing	
11	10	Mr.XYZ	Analyst	Software testing	
12	11	Mr.XYZ	Author	Software Reviewing	
13	12	Mr.XYZ	Analyst	Software testing	
14	13	Mr.XYZ	Author	Software Reviewing	
15	14	Mr.XYZ	Analyst	Software testing	

Otras características también fueron ampliadas, tales como el número máximo de hojas de cálculo que es posible crear por libro que pasó de 256 a 1.024 o la cantidad de memoria del PC que es posible emplear que creció

de 1 GB a 2 GB soportando además la posibilidad de usar procesadores de varios núcleos.

Algunos software GIS pueden abrir las tablas Microsoft Office Excel directamente y trabajar con ellas del mismo modo que trabaja con otras fuentes de datos tabulares.



Verifique si el software GIS que está utilizando puede (o no) abrir archivos de Microsoft Excel.

Bases de datos Access

Microsoft Access es una aplicación de bases de datos de Microsoft Office, dirigida a usuarios domésticos y pequeñas empresas.

Ofrece numerosas plantillas y una interfaz gráfica de usuario intuitiva que permite que cualquier usuario pueda aprender a manejarla rápidamente.

A diferencia de los formatos anteriores, y ya entrando en lo que se denomina **Segunda Generación de Bases de Datos**, la base de datos Access se apoya 100% en el modelo relacional y permite creación de múltiples bases de datos.

Tanto ArcGIS como QGIS permiten la lectura de datos temática a partir de estas bases de datos; esto en el ambiente Windows. Es posible crear y modificar datos temáticos desde el software que estamos utilizando.

Por otro lado, desde finales del siglo pasado existe un formato propietario de la empresa ESRI, llamado Personal Geodatabase, que extiende las bases de datos Access para almacenar datos geográficos.

Según la empresa ESRI, *“una geodatabase personal es una base de datos de Microsoft Access que puede almacenar, consultar y administrar tanto datos espaciales como datos no espaciales. Dado que se almacenan en bases de datos de Access, las geodatabases personales tienen un tamaño máximo de 2 GB. Además, solo una persona puede editar a la vez los datos de una geodatabase personal.”*. Dentro de una geodatabase personal podemos almacenar: datos vectoriales, ráster, tablas, anotaciones, TINs, etc.

A modo de ejemplo, la totalidad de los datos catastrales de nuestro país (versión Dirección Nacional de Catastro) se puede almacenar en una base de datos Access de 250MB.

A diferencia de los archivos SHP, las geodatabase personales no sirven para formato de intercambio ya que no todos los softwares permiten trabajar directamente con este formato.

Tareas que puede realizar con las tablas y la información de atributos

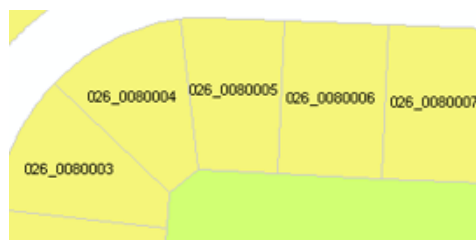
Hay muchas tareas de representación cartográfica, análisis y administración de datos que puede realizar con la componente temática, es decir, con los datos tabulares.

Las tablas permiten representar cartográficamente y visualizar los datos. Por ejemplo, podemos clasificar o categorizar atributos para simbolizar una capa.

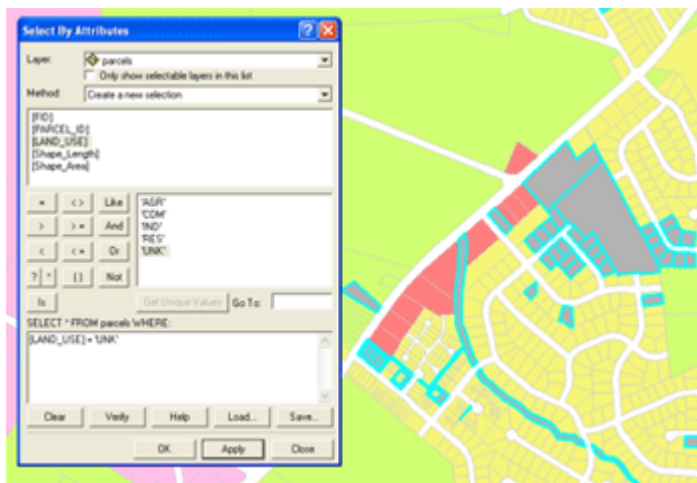
Podemos utilizar los valores de población para simbolizar las principales ciudades con un símbolo más grande que el que utilizaría para los pueblos y las poblaciones más pequeñas.

También podemos especificar que se utilizará un color diferente para representar cada tipo de uso del suelo en una capa de parcela.

Podemos utilizar los valores de atributo para generar texto para etiquetar cada entidad de parcela. En el siguiente gráfico, las parcelas están simbolizadas por el campo "tipo de uso del suelo", y luego están etiquetadas con los valores de "identificador de parcela". Esto lo veremos más adelante en el curso, durante la representación de los datos geográficos.



Los datos temáticos permiten realizar consultas y análisis espaciales. Esto lo veremos más adelante en este módulo. En el siguiente gráfico, se seleccionan las entidades con un valor LAND_USE de UNK (desconocido) mediante una consulta de atributos.



Cuando cambia la información de la realidad, podemos llevar adelante tareas de actualización de la componente temática.

Por ejemplo, deberemos actualizar la base de datos cuando el uso del suelo o el propietario de una parcela se modifican, o se clasifican los valores desconocidos. Si disponemos de una capa que representa algunas tuberías con un campo para el diámetro, fácilmente podemos cambiar los atributos cuando las cuadrillas de trabajo quita una tubería de 8 pulgadas y la reemplaza con una tubería de 6 pulgadas.

Consultas a partir de la componente temática

Las expresiones de consulta se usan en ArcGIS y QGIS para seleccionar un subconjunto de entidades y registros de tablas. Las expresiones de consulta en ArcGIS se ajustan a las expresiones **SQL** estándar.



SQL (por sus siglas en inglés Structured Query Language: lenguaje de consulta estructurada) es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales (DBMS) que permite especificar diversos tipos de operaciones en ellos. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas.

Podemos usar estas sintaxis mediante en las herramientas propias de la herramienta de cada software consultas para establecer una consulta de definición de capa.

Veremos en esta sección la manera de generar las expresiones de cláusula **WHERE** básicas.

Una expresión SQL simple

En SQL, la instrucción SELECT permite obtener un conjunto de registros.

Una instrucción SELECT recupera ninguna, una o más filas de una o más tablas de la base de datos.

Como SQL es un lenguaje de programación declarativo, las consultas SELECT especifican un conjunto de resultados, pero no especifican cómo calcularlo. El manejador de la base de datos traduce la consulta en un "plan de consulta" que puede variar entre ejecuciones, versiones de base de datos y software de base de datos. Esta funcionalidad se llama "optimizador de consultas" ya que es responsable de encontrar el mejor plan de ejecución posible para la consulta, dentro de las restricciones aplicables.

A modo de ejemplo: consideremos dada la siguiente relación:

```
ParcelasRurales (Departamento, Padrón, Superficie, ValorReal)
```

La siguiente consulta SQL permite recuperar todas las parcelas rurales.

```
Select * from ParcelasRurales
```

La siguiente consulta SQL permite recuperar todas las parcelas rurales del departamento de FLORES

```
Select * from ParcelasRurales where Departamento = 'FLORES'
```

La siguiente consulta SQL permite recuperar todas las parcelas rurales cuyo valor real es menor a 1,500.000\$

```
Select * from ParcelasRurales where ValorReal < 1500000
```

Como se puede todas las instrucciones de consulta comienza como **Select * from ParcelasRurales**. Esto lo podemos leer como: seleccionar (**Select**) todos los campos (*) de (**from**) la tabla ParcelasRurales.

Este tema se verá en extenso en el curso de **Diseño de Base de Datos Geográficas**.

Algunas consultas SQL pueden indicar la palabra **WHERE**; esta se especifica la condición de búsqueda de las filas devueltas por la consulta.

Una sentencia SQL al incluir el **WHERE** podría ser: **Select * from ParcelasRurales where Padron = 1400**. Esto lo podemos leer como: seleccionar (**Select**) todos los campos (*****) de (**from**) la tabla ParcelasRurales, donde (**WHERE**) el atributo Padrón es igual a 1400.

Consultas en los softwares GIS

En los software GIS, la parte "**SELECT * FROM**" forma la primera parte de la expresión SQL y se proporciona automáticamente, es decir, **no debemos ingresarla**. Se supone que cada software GIS lo incluye implícitamente. Por lo tanto, lo que deberemos incluir en las consultas es solamente lo que queda a la **DERECHA** de la palabra **WHERE**; esto es lo que entendemos como **condición de la búsqueda**.

La forma general para las expresiones simples de consulta en la mayoría de los software GIS queda como:

<Nombre_de campo> <Operador> <Valor o cadena de caracteres>

Decimos que son expresiones simples ya que cuenta con una sola secuencia de **nombre de campo**, **operador** y **valor o cadena de caracteres**.

Para las consultas compuestas, se utiliza la siguiente forma:

**<Nombre_de campo> <Operador> <Valor o cadena de caracteres> <Conector>
<Nombre_de campo> <Operador> <Valor o cadena de caracteres> ...**

Aquí tenemos varias secuencias de **nombre de campo**, **operador** y **valor o cadena de caracteres** relacionados a través de **conectores**. Los conectores que veremos en este curso son los operadores lógicos de **and** y **or**; esto lo veremos más adelante en este módulo.

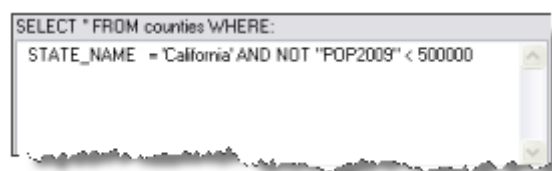
Un ejemplo a partir de la relación definida en las secciones anteriores sería.

Departamento = 'FLORES' and ValorReal > 300000

Esto lo entendemos como: Los departamentos de FLORES y cuyo valor real sea mayor a 300000.

En toda condición de búsqueda, opcionalmente, se pueden utilizar los paréntesis () para definir el orden de las operaciones en las consultas compuestas en forma similar al uso que le damos en las expresiones matemáticas.

En la mayoría de los cuadros de diálogo de los software GIS en los que genera una expresión de consulta, se le proporciona el nombre de la capa o tabla (o lo puede seleccionar desde una lista desplegable). A modo de ejemplo:



Buscar cadenas de caracteres

Las **cadenas de caracteres**, es decir los textos, siempre deben estar encerradas con comillas simples. Por ejemplo:

Departamento = 'FLORES'

En la mayoría de los DBMS, las cadenas de caracteres en las expresiones **distinguen entre mayúsculas y minúsculas**.

Para realizar una búsqueda que no distinga entre mayúsculas y minúsculas en otros formatos de datos, puede utilizar una función SQL para convertir todos los valores al mismo tipo de letra. Para esto es posible utilizar la función **UPPER** (mayúscula) o **LOWER** (minúscula).

Por ejemplo, la siguiente expresión seleccionará a las parcelas rurales cuyo departamento esté almacenado ya sea como Flores o como FLORES:

UPPER(Departamento) = 'FLORES'

Otros DBMS cuentan con funciones similares: **UCASE** y **LCASE** realizan la misma operación. **Será tarea de cada alumno determinar la sintaxis correcta según el software que utiliza y el manejador de base de datos.**

El operador LIKE (en lugar del operador =) permite llevar a cabo una búsqueda de cadena de caracteres parcial. Por ejemplo, esta expresión seleccionará FLORES y FLORIDA entre los nombres de los departamentos:

```
DEPARTAMENTO LIKE 'FLOR*'
```

El **asterisco (*)** o el **símbolo de porcentaje (%)** significa que se acepta cualquier elemento en su lugar: un carácter, cien caracteres o ningún carácter. El símbolo depende del DBMS.

Alternativamente, podemos buscar con **un comodín que representa un carácter**, usando el **infra guion (_)** o el **signo de interrogación**.

Por ejemplo, esta expresión serviría para buscar las personas cuya nacionalidad sea MEJICANO o MEXICANO:

```
NACIONALIDAD LIKE 'ME_ICANO'
```

Es posible utilizar los operadores **mayor** que (>), **menor** que (<), **mayor o igual** que (>=), **menor o igual** que (<=) y **BETWEEN** para seleccionar los valores de cadenas de caracteres basados en el ordenamiento.

Por ejemplo, esta expresión seleccionará todas las parcelas rurales de los departamentos cuyos nombres que comienzan con las letras de M a Z:

```
DEPARTAMENTO >= 'M'
```

También puede utilizar el operador **no igual** (<>) cuando realiza consultas de cadenas de caracteres. Por ejemplo, para seleccionar las parcelas rurales de todos los departamentos salvo el de RIVERA.

```
DEPARTAMENTO <> 'FLORES'
```

Existen **funciones de cadena de caracteres** que permiten para dar formato a las cadenas de caracteres. Por ejemplo, la función **LEFT** devuelve cierta cantidad de caracteres comenzando por la izquierda de la cadena de

caracteres. En este ejemplo, la consulta devolverá todos los estados que comiencen con la letra A:

```
LEFT (Departamento,1) = 'A'
```



Es fundamental consultar la documentación del DBMS que va a utilizar para obtener una lista de las funciones disponibles.

La palabra clave NULL

En páginas anteriores hemos presentado los valores nulos; los mismos se utilizan cuando no tenemos un valor para alguno de los atributos.

Es posible utilizar la palabra clave **NULL** para seleccionar las entidades y los registros que tienen valores nulos para el campo que se especificó. **La palabra clave NULL siempre está precedida por IS o IS NOT.**

Por ejemplo, para encontrar las parcelas rurales cuyos departamentos no fueron incluidos en la relación utilizada en las secciones anteriores.

```
Departamentos is null
```

Alternativamente, para encontrar las parcelas rurales para las que se introdujo el nombre de departamento, podemos utilizar

```
Departamento not is null
```

Buscar números

Es posible consultar números con los operadores **igual** (=), **no igual** (<>), **mayor que** (>), **menor que** (<), **mayor o igual que** (>=), **menor o igual que** (<=) y **BETWEEN**. A modo de ejemplo, la siguiente sentencia permite buscar las parcelas rurales cuyos valores reales sean menores a 100.000\$.

```
ValorReal < 100000
```



Los valores numéricos siempre se enumeran utilizando el punto como delimitador decimal independientemente de la configuración regional. La coma no se puede usar como delimitador decimal o de miles en una expresión.

Es posible utilizar utilizar las funciones numéricas para dar formato a los números. Por ejemplo, la función **ROUND** redondeará un número a una cantidad de decimales:

ROUND(Superficie,0) = 50000



Es fundamental consultar la documentación del DBMS para obtener una lista de las funciones compatibles.

Combinar expresiones

Es posible generar expresiones complejas combinando las expresiones con los operadores lógicos **AND** y **OR**.

Por ejemplo, la siguiente expresión selecciona todas las parcelas rurales del departamento de Durazno cuya superficie sea menor a 5 hectáreas.

Departamento = 'DURAZNO' and ValorReal < 50000

Cuando utilizamos el operador **OR**, al menos una de las dos partes de la expresión separada por el operador **OR** debe ser verdadera para que se seleccione el registro.

Utilizaremos el operador **NOT** al comienzo de una expresión para buscar entidades o registros que no coincidan con la expresión especificada.

Por ejemplo, una forma de identificar las parcelas rurales que no pertenecen al departamento de Rivera, podría ser:

not Departamento = 'RIVERA'

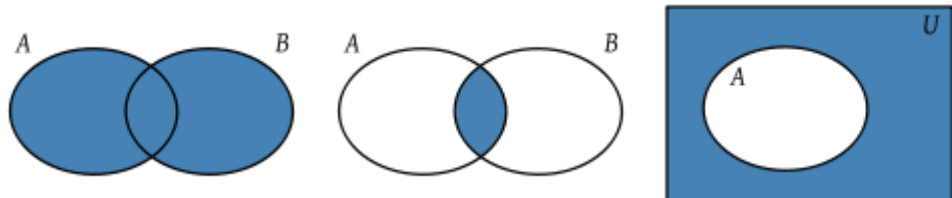
Las expresiones con **NOT** se pueden combinar con **AND** y **OR**.

El operador lógico AND nos devuelve un valor verdadero, si ambas expresiones que conecta son verdaderas. El operador lógico OR nos devuelve un valor verdadero, si por lo menos una de las expresiones es verdadera. El operador lógico NOT, nos invierte el valor de la expresión. Esto se sintetiza en el siguiente cuadro.

Exp1	Exp2	Exp1 and Exp2	Exp1 or Exp2
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Exp1	Not Exp1
True	False
False	True

Estos operadores se basan en lo conocemos como algebra de conjuntos; que estudia las operaciones básicas que pueden realizarse con conjuntos, como la unión, intersección y complementación. La unión está asociada conceptualmente al operador lógico **OR**, al intersección al **AND** y el complemento al **NOT**.



Operadores aritméticos

Se utiliza un operador aritmético para sumar, restar, multiplicar y dividir valores numéricos.

Operador	Descripción
*	Operador aritmético para la multiplicación .
/	Operador aritmético para la división .
+	Operador aritmético para la suma .
-	Operador aritmético para la resta .

A modo de ejemplo, a partir de la siguiente relación:

ParcelaUrbana (Depto, Padron, SupConstruida, SupSinConstruir)

La siguiente expresión permite seleccionar las parcelas urbanas cuya superficie total (es decir, la suma de superficie construida y de la superficie sin construir) es menor o igual a 1000 metros cuadrados.

SupConstruida + SupSinConstruir < 1000

La siguiente expresión permite seleccionar las parcelas urbanas del departamento de Flores, y que cuya superficie construida es menor al 10% que la superficie sin construir.

(Depto = 'FLORES') and (SupConstruida < SupSinConstruir / 10)



Operadores de comparación

Veremos ahora en detalle los operadores de comparación, estos utilizan operadores de comparación para comparar una expresión con otra.

Operador	Descripción
<	Menor que. Se puede usar con cadenas (la comparación está basada en un orden alfabético), números y fechas.
<=	Menor o igual que. Se puede usar con cadenas (la comparación está basada en un orden alfabético), números y fechas.
<>	No igual que. Se puede usar con cadenas (la comparación está basada en un orden alfabético), números y fechas.
>	Mayor que. Se puede usar con cadenas (la comparación está basada en un orden alfabético), números y fechas.
>=	Mayor o igual que. Se puede usar con cadenas (la comparación está basada en un orden alfabético), números y fechas. Por ejemplo, esta consulta selecciona todas las ciudades con nombres que comienzan con las letras de M a Z: <pre>"CITY_NAME" >= 'M'</pre>
[NOT] BETWEEN x AND y	Selecciona un registro si tiene un valor mayor o igual que x y menor o igual que y . Cuando está precedido por NOT, selecciona un registro si tiene un valor fuera del rango especificado. Por ejemplo, esta expresión selecciona todos los registros con un valor mayor o igual que 1 y menor o igual que 10: <pre>"OBJECTID" BETWEEN 1 AND 10</pre> Este es el equivalente de la siguiente expresión: <pre>"OBJECTID" >= 1 AND OBJECTID <= 10</pre>
IS [NOT] NULL	Selecciona un registro si tiene un valor nulo para el campo especificado. Cuando NULL está precedido por NOT, selecciona un registro si tiene algún valor para el campo especificado. Por ejemplo, esta expresión selecciona todos los registros con un valor nulo para la población: <pre>"POPULATION" IS NULL</pre>
x [NOT] LIKE	Utilice el operador LIKE (en lugar del operador =) con comodines para crear una búsqueda de cadena de caracteres parcial. Por ejemplo, esta expresión selecciona Mississippi y Missouri entre los

nombres de estados de EE.UU.:

```
"STATE_NAME" LIKE 'Miss%'
```

El símbolo de porcentaje (%) significa que se acepta cualquier elemento en su lugar: un carácter, cien caracteres o ningún carácter. Alternativamente, si desea buscar con un comodín que represente un carácter, utilice un guión bajo (_). Por ejemplo, esta expresión encuentra Catherine Smith y Katherine Smith:

```
"OWNER_NAME" LIKE '_atherine Smith'
```

Los comodines con símbolo de porcentaje y guión bajo funcionan en cualquier tipo de datos basados en archivos o datos de geodatabase multiusuario. LIKE funciona con datos de carácter en ambos lados de la expresión.

Operadores lógicos

Operador	Descripción
AND	Combina dos condiciones juntas y selecciona un registro si las dos condiciones son verdaderas.
OR	Combina dos condiciones juntas y selecciona un registro si al menos una condición es verdadera.
NOT	Selecciona un registro si no coincide con la expresión.



Funciones

A continuación se muestra la lista completa de funciones admitidas por algunas de los manejadores de bases de datos más utilizados.

Funciones de cadena

Los argumentos denotados como *cadena_exp* pueden ser el nombre de una columna, una cadena de caracteres literal o el resultado de otra función, donde el tipo de datos subyacentes se pueda representar como un tipo de carácter.

Función	Descripción
CHAR_LENGTH (<i>cadena</i>)	Devuelve la longitud en caracteres de la expresión de <i>cadena</i> .
LOWER (<i>cadena</i>)	Devuelve una cadena de caracteres equivalente a la de <i>cadena</i> , con todos los caracteres en mayúsculas convertidos en minúsculas.
TRIM (<i>cadena</i>)	Devuelve la <i>cadena</i> quitando los espacios que sobran del extremo inicial y del extremo final o de ambos extremos de la <i>cadena</i> .
UPPER (<i>cadena</i>)	Devuelve una cadena equivalente a la de <i>cadena</i> , con todos los caracteres en minúsculas convertidos en mayúsculas.

Funciones numéricas

Todas las funciones numéricas devuelven un valor numérico.

Función	Descripción
ABS (<i>número</i>)	Devuelve el valor absoluto de <i>número</i> .
ACOS (<i>número</i>)	Devuelve el arcocoseno del <i>número</i> como un ángulo expresado en radianes.
ASIN (<i>número</i>)	Devuelve el arcoseno del <i>número</i> como un ángulo

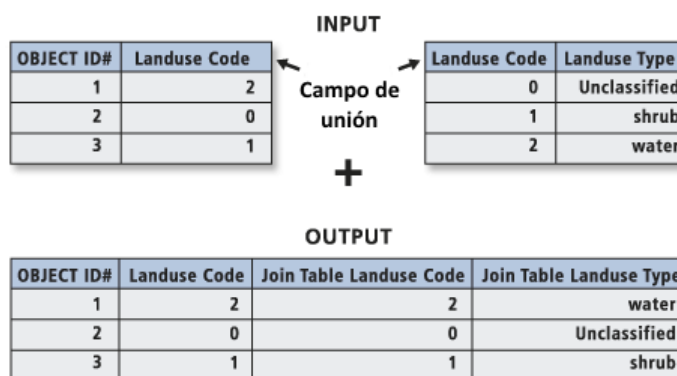
	expresado en radianes.
ATAN (<i>número</i>)	Devuelve el arco tangente de <i>número</i> como un ángulo expresado en radianes.
CEILING (<i>número</i>)	Devuelve el entero más pequeño mayor o igual que <i>número</i> .
COS (<i>número</i>)	Devuelve el coseno de <i>número</i> , donde <i>número</i> es un ángulo expresado en radianes.
FLOOR (<i>número</i>)	Devuelve el entero más grande menor o igual que <i>número</i> .
LOG (<i>número</i>)	Devuelve el logaritmo neperiano de <i>número</i> .
LOG10 (<i>número</i>)	Devuelve el logaritmo de base 10 de <i>número</i> .
MOD (<i>número1</i> , <i>número2</i>)	Devuelve el resto de <i>número1</i> dividido por <i>número2</i> .
POWER (<i>número1</i> , <i>número2</i>)	Devuelve el valor de <i>número1</i> la potencia de <i>número2</i> .
ROUND (<i>número1</i> , <i>número2</i>)	Devuelve <i>número1</i> redondeado a <i>número2</i> lugares a la derecha del punto decimal.
SIGN (<i>número</i>)	Devuelve un indicador del signo de <i>número</i> . Si <i>número</i> es menor que cero, se devuelve -1. Si <i>número</i> es igual a cero; se devuelve 0. Si <i>número</i> es mayor que cero; se devuelve 1.
SIN (<i>número</i>)	Devuelve el seno de <i>número</i> , donde <i>número</i> es un ángulo expresado en radianes.
TAN (<i>número</i>)	Devuelve la tangente de <i>número</i> , donde <i>número</i> es un ángulo expresado en radianes.
TRUNCATE (<i>número1</i> , <i>número2</i>)	Devuelve <i>número1</i> truncado a <i>número2</i> lugares a la derecha del punto decimal.

Llegando al final: la unión de tablas (JOINS)

Como ya hemos visto y comentando ampliamente, las bases de datos modernas promueven su organización en varias tablas, **cada una enfocada en un tema específico, en lugar de una tabla grande que contenga todos los campos necesarios.**

Contar con varias tablas evita la duplicación de información en la base de datos, debido a que almacena la información sólo una vez en la tabla. Cuando necesita información que no se encuentra en la tabla actual, puede unir temporalmente las dos tablas.

A modo de ejemplo, podemos obtener datos de otros departamentos de la organización, adquirir datos disponibles comercialmente o descargar datos de Internet. Si esta información se almacena en una tabla, como una tabla de dBase, CSV o EXCEL, podremos asociarla a sus entidades geográficas y mostrar los datos en el mapa.



Los softwares GIS nos permiten asociar registros en una tabla con registros en otra tabla a través de un campo común. Podemos **unir de manera temporal** en el mapa.

Al unir dos tablas, **se anexan los atributos de una en la otra en función de un campo común a ambas.**

Generalmente, uniremos una tabla de datos a una capa a partir del valor de un campo existente en ambas tablas. **El nombre del campo no tiene que ser el mismo, aunque sí el tipo de datos; podemos unir números a números, cadenas de caracteres a cadenas de caracteres, etc.**

Relaciones de uno a uno.

Al unir tablas, se establece una relación de uno a uno o de muchos a uno entre la tabla de atributo de la capa y la tabla que contiene la información que se desea unir.

A modo de ejemplo, se muestra una relación de uno a uno entre cada condado y los datos de cambio de población de dicho condado. Es decir, hay un dato de población concreto para cada condado.

The diagram illustrates a one-to-one relationship between two tables: 'US Counties' and 'Population_Data'. A blue arrow labeled 'One to One' points from the 'ID' column of the 'US Counties' table to the 'ID' column of the 'Population_Data' table. Both columns have the value '16033' highlighted with a blue box.

US Counties		
FID *	Name	ID
190	Canyon County	16027
256	Caribou County	16029
208	Cassia County	16031
225	Clark County	16033
226	Clearwater County	16035
227	Custer County	16037
228	Elmore County	16039

Population_Data	
ID	% Population Change
16027	+3.9
16029	+0.5
16031	+0.9
16033	+3
16035	+0.5
16037	+0.5
16039	+3.2

Relaciones de uno a muchos.

Por otro lado, supongamos que tenemos una capa donde cada polígono está clasificado según el tipo de uso del suelo. La tabla de atributos de la capa solo almacena un código del uso del suelo; en una tabla independiente se almacena la descripción completa de cada tipo de uso del suelo. Al unir estas dos tablas, se establece una relación de muchos a uno porque muchos registros de la tabla de atributo de la capa se unen al mismo registro de la tabla de descripciones del uso del suelo. A continuación, podríamos utilizar un texto más descriptivo a la hora de generar la leyenda del mapa.

The diagram illustrates a many-to-one relationship between two tables: 'Landuse' and 'Zone_Code'. A blue arrow points from the 'ZONE_CODE' column of the 'Landuse' table to the 'ZONE_CODE' column of the 'Zone_Code' table. The 'Landuse' table has five rows with 'ZONE_CODE' values 'MDR', 'VAC', 'LDR', 'MDR', and 'MDR'. The 'Zone_Code' table has four rows with 'ZONE_CODE' values 'SDP', 'TNS', 'MDR', 'LDR', and 'VAC'.

Landuse		
OBJECTID *	Shape *	ZONE_CODE
384	Polygon	MDR
385	Polygon	VAC
386	Polygon	LDR
387	Polygon	MDR
388	Polygon	MDR

Zone_Code		
FID *	ZONE_CODE	DESCRIPTION
10	SDP	Special Development Plan
11	TNS	Transitional
12	MDR	Medium-density Residential
13	LDR	Low-density Residential
14	VAC	Vacant

Pasos básicos

La operación que permite la unión de dos tablas de atributos requiere conocer las características de las tablas que se desean unir, y la dirección de la unión, es decir:

- Cuál es el campo que posee valores similares en cada tabla; es decir, **el campo común que permitirá asociar los registros de una y otra tabla garantizando su integridad.**
- Cuál es la **tabla que va a aumentar el número de campos gracias a la anexión de otra tabla.**

Ambas cuestiones son claves porque si no existe un campo con valores comunes en cada tabla, estas no podrán unirse.

Ejercicios de consulta

Para terminar este módulo dejaremos planteados algunos ejercicios a efectos que puedan ser utilizados por los estudiantes.

Ejercicio del mismo tipo podrá ser propuesto en el próximo parcial. El mismo no será resuelto en clase ni forma parte de ninguna entrega; no obstante esto, el estudiante podrá entregárselo al docente para su corrección.

Estaciones meteorológicas móviles

Supongamos que tenemos posibilidad de diseñar una tabla para almacenar los datos de 50 estaciones meteorológicas móviles a lo largo del tiempo. El instrumental técnico que pueden disponer las estaciones son:



- **Termómetro:** Instrumento que mide la temperatura en diversas horas del día.
- **Geotermómetro:** Para medir la temperatura a 50 cm de profundidad.
- **Termómetro de mínima junto al suelo:** Mide la temperatura mínima a una distancia de 15 cm sobre el suelo.
- **Barómetro:** Mide la presión atmosférica en la superficie.
- **Pluviómetro:** Mide la cantidad de agua caída sobre el suelo por metro cuadrado en forma de lluvia, nieve o granizo.

- **Psicrómetro:** Medida de la humedad relativa del aire.
- **Piranómetro:** Medida de la radiación solar global.
- **Heliógrafo:** Medida de las horas de luz solar.
- **Anemómetro:** Medida de la velocidad del viento.
- **Veleta:** Instrumento que indica la dirección del viento.
- **Nefobasímetro:** Medida de la altura de las nubes, pero sólo en el punto donde éste se encuentre colocado.
- **Dispositivo GNSS:** Para determinar la Latitud y Longitud del equipamiento.
- **Reloj;** Para determinar la hora de captura de los datos.

No todas las estaciones tienen todo el instrumental, por lo tanto existe la posibilidad de no contar con todos los datos para todas las estaciones móviles. Cada estación móvil estará durante una hora recolectando la información; y se almacenará los datos una única vez cada hora. Se supone que las estaciones móviles trabajan 12 horas por día, durante seis días a la semana.

Cada estación está identificada por un valor numérico único; cada estación está siendo operada por una persona que está identificada por su número de cédula.

Tener en cuenta, para el diseño de la tabla, que será necesario responder a las siguientes consultas:

- Registros con temperatura geotérmica mayor a 16 grados, a las 12 del mediodía.
- Registros con altura de las nubes menor a 600 metros, cuando la velocidad del viento es mayor a 20Km/h.
- Registros cuya diferencia entre la temperatura del aire y la temperatura a 15cm del suelo sea 3 grados (en valor absoluto), luego de las 13 horas.
- Registros que no tienen datos de medida de presión atmosférica en la superficie.
- Registros que fueron recolectados por el operador cuya cédula de identidad es '2,561.111-0'.
- Registros con están fuera de la extensión geográfica del Uruguay.
- Registros cuya parte entera de la temperatura sea menor a 10 grados.
- Registros que están a 100 kms de Montevideo y fueron recolectados antes de las 10:00.

El estudiante deberá definir la relación (es decir la tabla con sus atributos y tipos de datos) que permita almacenar toda la información recolectada por la estación móvil y el operador de la misma.

¿Qué cardinalidad tendrá toda la tabla a las 52 semanas de comenzada la recolección de datos?

¿Qué tipo de geometría sería la más conveniente de utilizar?

Escribir las consultas planteadas anteriormente.