



ING. MARCELO MORA
2020

Repaso General



Conceptos

- ❑ ¿Qué es Dato?
- ❑ ¿Qué es Información?
- ❑ ¿Qué es un Campo?
- ❑ ¿Qué es un Registro?
- ❑ ¿Qué es un Archivo?
- ❑ ¿Qué es una Base de Datos?
- ❑ ¿Qué es un Objeto?
- ❑ ¿Qué es el Lenguaje Procedimental y No Procedimental?
- ❑ ¿Qué es el DML – el DDL – el DCL ?

Conceptos

- ❑ **Dato** = es el elemento primario del concepto “Información”, es una palabra en singular que se refiere a un hecho.
 - Se puede mencionar que es la representación convencional de una unidad mínima de información.
 - ✓ Ejemplos: Azul , F, M, Activo, Verdadero, Juan...
- ❑ **Información** = es un conjunto organizado de datos que constituye un mensaje sobre un cierto fenómeno, ente o persona, lo cual nos puede ayudar a resolver problemas y tomar decisiones, ya que su uso racional es en base al conocimiento.

Conceptos

- ❑ **Campo** = es una columna o pieza de información.
- ❑ **Registro o Tupla** = es un sistema o conjunto completo de campos, con sus respectivos datos.
- ❑ **Archivo o Fichero** = es una colección o conjunto organizado de registros.
 - Ejemplo: Una agenda telefónica es un archivo, esto ya que contiene una lista de registros, cada uno de los cuales consiste en varios campos como:
nombre, apellido_1, apellido_2,
dirección, número de teléfono,
número de celular, etc...

Conceptos

- ❑ **Base de Datos (BD)** = Técnicamente se puede ver como un almacén de datos relacionados con diferentes modos de organización, diseñado para representar algunos aspectos del mundo real, de interés al diseñador de la mismo.
 - ✓ Es una entidad administrada, en la cual se pueden almacenar datos de manera estructurada y relacionada, con la menor redundancia posible para que diferentes programas y usuarios puedan utilizar estos datos.
 - ✓ Una base de datos es un sistema de archivos o ficheros electrónicos.

Sentencias del SQL

- ❑ Las sentencias en SQL se dividen principalmente en las siguientes categorías:
 - **DDL** (Data Definition Language – Lenguaje de Definición de Datos): es un lenguaje proporcionado por el SGBD, que permite a los usuarios llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.
 - **DML** (Data Manipulation Language – Lenguaje de Manipulación de Datos): es un lenguaje proporcionado por el SGBD que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado, esto mediante la generación de consultas para ordenar, filtrar y extraer datos de la base de datos.
 - **DCL** (Data Control Language – Lenguaje de Control de Datos): es un lenguaje proporcionado por el SGBD que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la BD.

Instancias y Esquemas de una BD

- Con el paso del tiempo la información que se va acumulando y desechando en la base de datos, ocasiona que ésta cambie, lo que da el origen a las instancias y a los esquemas de BD's.

➤ Instancia

- Se denomina instancia al estado que presenta una base de datos en un tiempo dado.
- Veámoslo como una fotografía que tomamos de la base de datos en un tiempo t , después de que transcurre el tiempo t la base de datos ya no es la misma.

Instancias y Esquemas de una BD

➤ Esquema

- Es la descripción lógica de la base de datos, encargada de proporcionar los nombres de las entidades y sus atributos especificando las relaciones que existen entre ellos.
- Es un banco en el que se inscriben los valores que irán formando cada uno de los atributos. El esquema no cambia los que varían son los datos y con esto tenemos una nueva instancia.

Instancias y Esquemas de una BD

- Considerando el ejemplo del vendedor que vende artículos, esquema e instancia según nuestro ejemplo, quedaría:

- Esquema:

- { Vendedor : Nombre, Puesto, Salario, RFC }
- { Artículo : Clave, Costo, Descripción }

- Instancia:

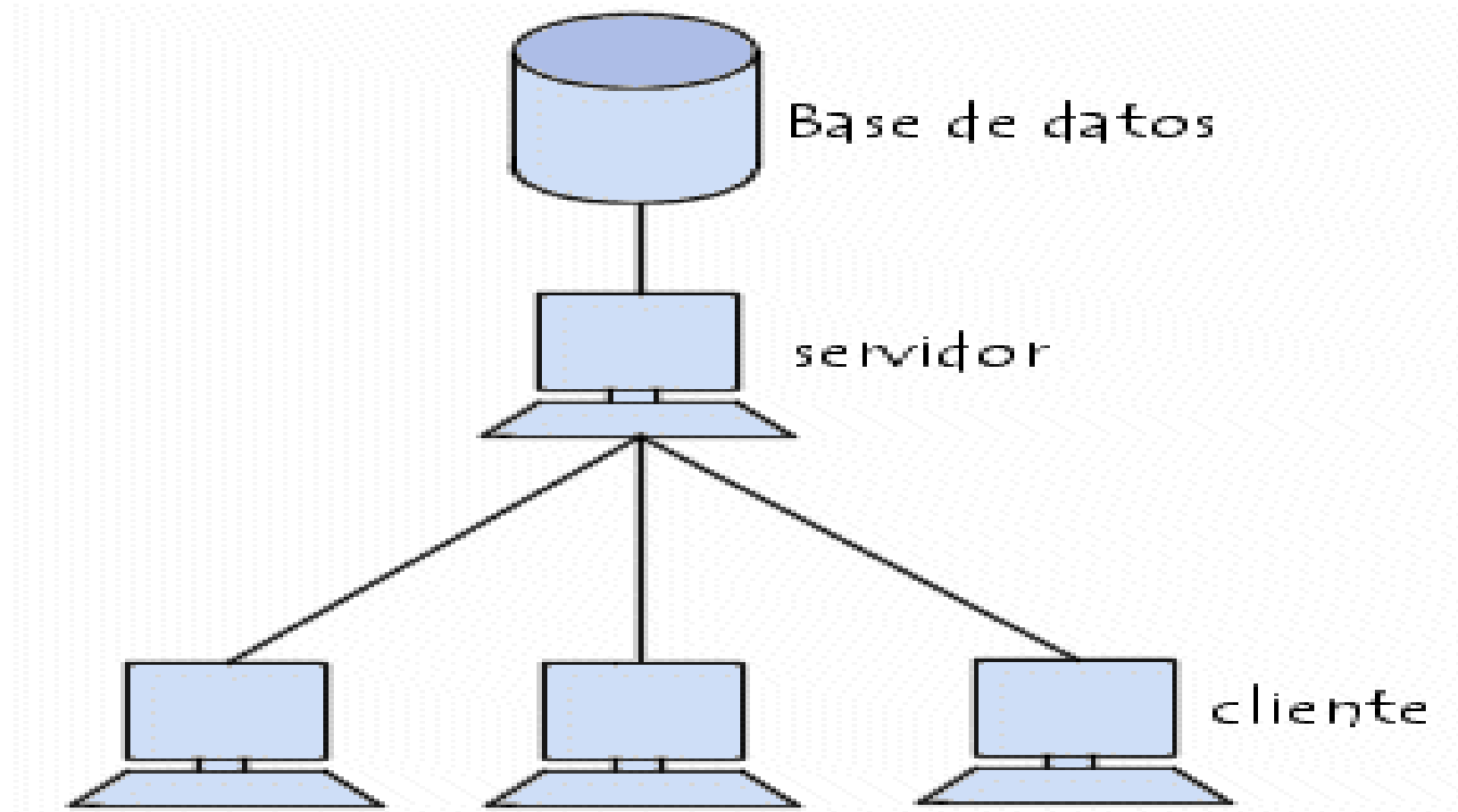
Juan Pérez Cota	Vendedor	5,000	PECJ500922XYZ	C001	250	Colcha matrimonial
-----------------	----------	-------	---------------	------	-----	--------------------

- Como podemos observar el esquema nos muestra la estructura en el cual se almacenaran los datos, en este caso en registros cuyos nombres de campos son: por parte del vendedor (Nombre, Puesto, Salario, RFC) y por el artículo (Clave, Costo, Descripción).
- La instancia representa a una serie de datos almacenados en los registros establecidos por el esquema, estos datos varían, no permanecen fijos en el tiempo.

Independencia de los datos

- Cuando hablamos de termino “**Independencia de Datos**”, nos referimos a la protección contra los programas o aplicaciones, los cuales puedan originar modificaciones cuando se altera la organización física o lógica de la base de datos.
- Existen 2 niveles de independencia de datos.
 - **Independencia Física de Datos:**
 - Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación.
 - **Independencia Lógica de Datos:**
 - Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación.

Estructura Básica de las BD



Preguntas ???...

Supuestos Iniciales: “Tipos de Datos”

- ❑ SQL en la actualidad admite una variada gama de tipos de datos para el tratamiento de la información contenida en las tablas, los tipos de datos pueden ser:
 - Numéricos (con o sin decimales)
 - Alfanuméricos,
 - De Fecha
 - Booleanos(si o no).

- ❑ Según el gestor de base de datos que estemos utilizando los tipos de datos varían, pero se reducen básicamente a los expuestos anteriormente, aunque en la actualidad casi todos los gestores de bases de datos soportan un nuevo tipo que es el tipo **BLOB (Binary Large Object)**, que es un tipo de datos especial destinado a almacenar archivos, imágenes (Datos multimedia).

Supuestos Iniciales: “Tipos de Datos”

Tipos de datos numéricos		
Tipo	Definición	Bytes
Integer	Valores enteros con signo.	4
Numeric(n,m)	Números reales de hasta 18 dígitos (con decimales), donde n representa el total de dígitos admitidos (normalmente denominado precisión) y m el número de posiciones decimales (escala).	5-17
Decimal(n,m)	Igual que el tipo numeric.	5-17
Float	Número de coma flotante, este tipo de datos se suele utilizar para los valores en notación científica.	4-8
Tipos de datos alfanuméricos		
Tipo	Definición	Bytes
char(n)	Almacena de 1 a 255 caracteres alfanuméricos. Este valor viene dado por n, y es el tamaño utilizado en disco para almacenar dato. Es decir si defino un campo como char(255), el tamaño real del campo será de 255, aunque el valor solo contenga 100.	0-255
varchar(n)	Igual que el tipo char, con la salvedad que varchar almacena únicamente los bytes que contenga el valor del campo.	0-255
Nota:El tamaño del campo varia en función de cada base de datos, siendo 255 el valor standart. En realidad el tamaño viene delimitado por el tamaño de las páginas de datos, para SQL Server el límite esta en 8000 bytes (8000 caracteres), siempre y cuando tengamos definido el tamaño de la página de datos a 8K		

Supuestos Iniciales: “Tipos de Datos”

Tipos de datos fecha		
Tipo	Definición	Bytes
Date	Almacena fechas, con día, mes y año.	8
Datetime	Almacena fechas con fecha y hora	4
Nota:La aparición de los tipos de datos de fecha supuso una atenta revolución el mundo de la bases de datos, en realidad, la base de datos almacena internamente números enteros, de hay que el tamaño sea de 4 bytes y 8 bytes (2 enteros), pero aporta la validación del dato introducido.		
Tipos de datos lógicos		
Tipo	Definición	Bytes
Bit	Tipo bit. Almacena un 0 ó no cero, según las bases de datos será 1 ó -1. Se aplica la lógica booleana, 0 es falso y no cero verdadero.	1 bit
Tipos de datos BLOB		
Tipo	Definición	Bytes
Image	Almacena imágenes en formato binario, hasta un máximo de 2 Gb de tamaño.	0-2Gb
Text	Almacena texto en formato binario, hasta un máximo de 2 Gb de tamaño.	0-2Gb

Supuestos Iniciales: “Operadores”

- ❑ Los operadores se pueden definir como combinaciones de caracteres que se utilizan tanto para realizar asignaciones como comparaciones entre datos.
- ❑ Los operadores se dividen en
 - ✓ **Aritméticos**
 - ✓ **Relacionales**
 - ✓ **Lógicos**
 - ✓ **De Concatenación**

Supuestos Iniciales: “Operadores”

Operadores SQL		
Aritméticos	+	Suma
	-	Resta
	*	Producto
	/	División
	** ^	Exponenciación
Relacionales	<	Menor que
	<=	Menor o igual que
	>	Mayor que
	>=	Mayor o igual que
	<> !=	Distinto
	!<	No menor que
	!>	No mayor que
Lógicos	AND	Los operadores lógicos permiten comparar expresiones lógicas devolviendo siempre un valor verdadero o falso. Los operadores lógicos se evalúan de izquierda a derecha.
	OR	
	NOT	
Concatenación	+	Se emplea para unir datos de tipo alfanumérico.

Supuestos Iniciales: “Palabras Claves o Reservadas”

- ❑ Las palabras clave son identificadores con un significado especial para SQL, por lo que no pueden ser utilizadas para otro propósito distinto al que han sido pensadas.
- ❑ SQL dispone de muy pocas órdenes, pero de múltiples palabras clave, lo que le convierten en un lenguaje sencillo pero tremendamente potente para llevar a cabo su función.

Palabras Clave			
ALL	AND	ANY	ASC
AVG	BEGIN	BY	CHAR
CHECK	CLOSE	COUNT	COMMIT
CREATE	CURSOR	DECIMAL	DECLARE
DELETE	DESC	DISTINCT	DEFAULT
EXISTS	FETCH	FLOAT	FOR
FROM	GRANT	GROUP	HAVING
IN	INDEX	INSERT	INTEGER
INTO	LIKE	MAX	MIN
NOT	NUMERIC	ON	OPEN
OR	ORDER	REVOKE	ROLLBACK
SELECT	SET	SUM	TABLE
UNION	UNIQUE	UPDATE	USER
VALUES	VIEW	WHERE	WITH

Supuestos Iniciales: “Funciones Agregadas”

- ❑ Las funciones agregadas proporcionan a SQL utilidades de cálculo sobre los datos de las tablas.
- ❑ Estas funciones se incorporan en las consultas SELECT y retornan un único valor al operar sobre un grupo de registros.
- ❑ Las funciones agregadas son.

Funciones Agregadas	
MAX()	Devuelve el valor máximo.
MIN()	Devuelve el valor mínimo.
SUM()	Devuelve el valor de la suma de los valores del campo.
COUNT()	Devuelve el número de filas que cumplen la condición
AVG()	Devuelve el promedio de los valores del campo

Supuestos Iniciales: “Predicados”

- ❑ Los predicados son condiciones que se indican en clausula WHERE de una consulta SQL.
- ❑ La siguiente tabla ilustra los predicados de SQL.

Predicados SQL	
BETWEEN...AND	Comprueba que al valor esta dentro de un intervalo
LIKE	Compara un campo con una cadena alfanumérica. LIKE admite el uso de caracteres comodines
ALL	Señala a todos los elementos de la selección de la consulta
ANY	Indica que la condición se cumplirá si la comparación es cierta para al menos un elemento del conjunto.
EXISTS	Devuelve un valor verdadero si el resultado de una subconsulta devuelve resultados.
IN	Comprueba si un campo se encuentra dentro de un determinado rango. El rango puede ser una sentencia SELECT.

Preguntas ???...

Supuestos Iniciales: “ Variables en SQL”

- ❑ Una variable es un valor identificado por un nombre (identificador) sobre el que podemos realizar modificaciones.
- ❑ En SQL los identificadores de variables deben comenzar por el carácter @, es decir, el nombre de una variable debe comenzar por @.
- ❑ Para declarar variables en SQL debemos utilizar la palabra clave **Declare**, seguido del identificador y tipo de datos de la variable.

```
-- Esto es un comentario de linea simple
/*
Este es un comentario con varias líneas.
Conjunto de Lineas.
*/

declare @nombre varchar(50) -- declare declara una variable
                                -- @nombre es el identificador de la
                                -- variable de tipo varchar
set @nombre = 'www.devjoker.com' -- El signo = es un operador
                                -- www.devjoker.com es un literal
print @Nombre -- Imprime por pantalla el valor de @nombre.
                -- No diferencia mayúsculas ni minúsculas
```

Supuestos Iniciales: “ Variables en SQL”

- ❑ Una variable es un valor identificado por un nombre (identificador) sobre el que podemos realizar modificaciones.
- ❑ En SQL podemos asignar valores a una variable de varias formas:
 - A través de la instrucción **SET**.
 - Utilizando una sentencia **SELECT**.
- ❑ El siguiente ejemplo muestra como asignar una variable utilizando la instrucción SET.

```
DECLARE @nombre VARCHAR(100)
-- La consulta debe devolver un único registro
SET @nombre = (SELECT nombre
                FROM CLIENTES
                WHERE ID = 1)
PRINT @nombre
```


Supuestos Iniciales: “ Variables en SQL”

El siguiente ejemplo muestra como asignar variables utilizando una sentencia SELECT.

```
DECLARE @nombre VARCHAR(100),
        @apellido1 VARCHAR(100),
        @apellido2 VARCHAR(100)

SELECT  @nombre=nombre ,
        @apellido1=Apellido1,
        @apellido2=Apellido2
FROM CLIENTES
WHERE ID = 1

PRINT @nombre
PRINT @apellido1
PRINT @apellido2
```