Laboratorio IV – SQL

SQL (*Structured Query Language*)

Ing. Gregorio Nicolás Tkachuk

Objetivos

- Que el estudiante logre:
  - Conocer y utilizar los comandos de SQL.
  - Crear y modificar Bases de Datos relacionales a través de los comandos específicos.

SQL: Contenidos

- SQL (Structured Query Language)
- SGBD (Sistema de Gestión de Base de Datos relacional)
- Lenguaje SQL, grupos de sentencias
  - DML (*Data Manipulation Language*)
  - DDL (*Data Definition Language*)
  - DCL (*Data Control Language*)
- Base de Datos, Modelo catálogo/esquema
- SQL Componentes de una Sentencia
- Precedencia de Operadores
- Tipos de Datos
- DDL (Create, Drop, Alter, Rename)

Que es el SQL (Structured Query Language)

- El SQL es un Lenguaje de Consulta Estructurado que permite expresar diversas operaciones, como por ejemplo, aritméticas, combinatorias, lógicas, selección y ordenación con datos almacenados en una Base de Datos Relacional.

- Base de Datos Relacional son aquellas que se caracterizan porque la información esta contenida en estructuras, llamadas tablas, donde los datos están dispuestos en filas y columnas.

Sistema de Gestión de Base de Datos Relacionales

- **SGBD** (Sistema de Gestión de Base de Datos relacional) o **RDBMS** (del inglés *Relational Database Management System*). Es el software exclusivamente dedicado a tratar con bases de datos relacionales.
- SGBD que soportan SQL: MySQL, PostgreSQL, Oracle, DB2, Microsoft SQL Server.

Funciones de un SGBD

Funciones SGBD

- Consulta y actualización de datos
- Mantenimiento de esquemas
- Manejo de transacciones

Sistema de Gestión de Base de Datos Relacionales

- SGBD que soportan SQL:
- **Oracle** (Oracle)
- **DB2** (IBM)
- **SQL Server** (Microsoft)
- **Sybase ASE** (Sybase)
- **Informix** (IBM)

Sistema de Gestión de Base de Datos Relacionales

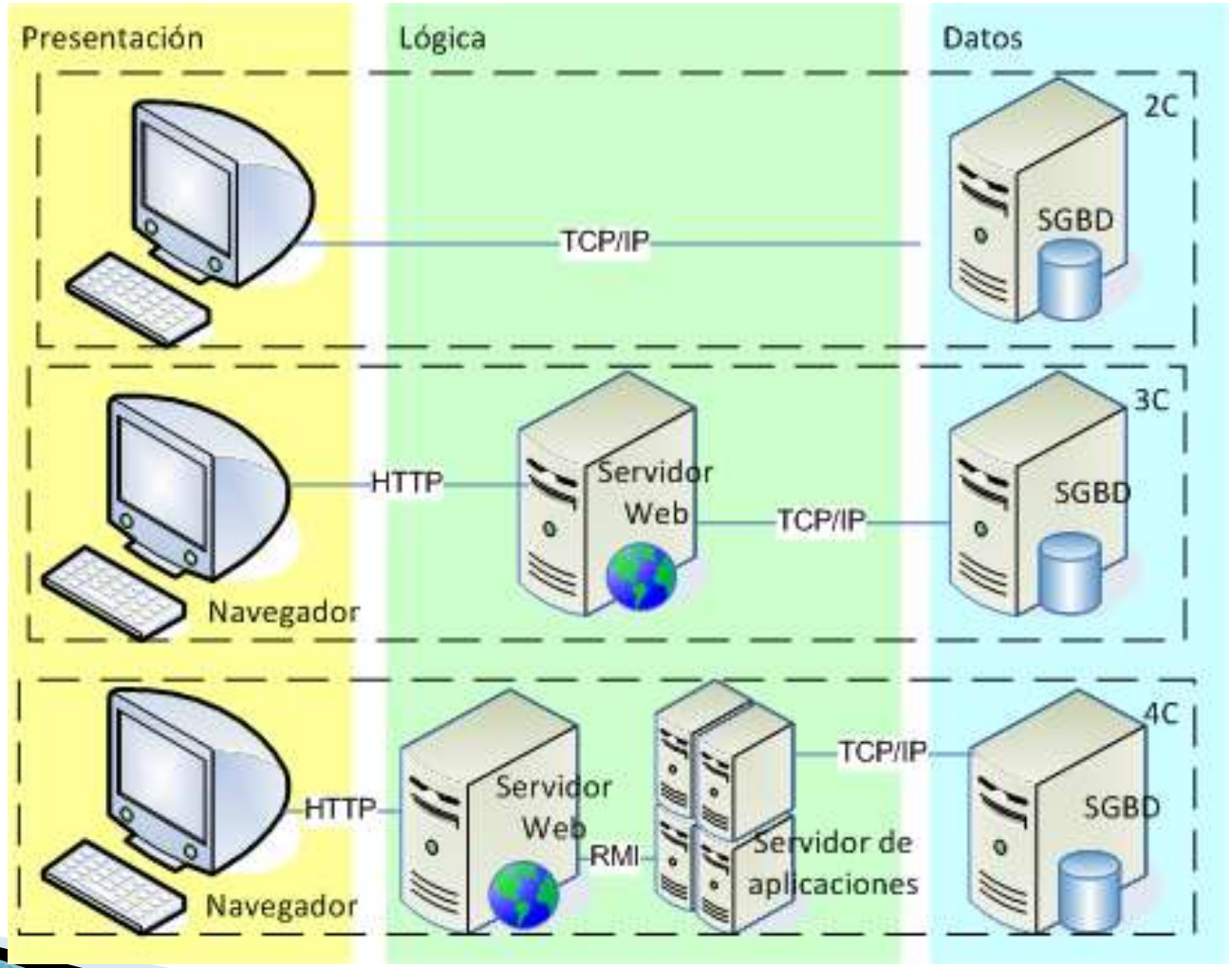
- SGBD que soportan SQL:

- **MySQL** (libre, licencia GPL, Oracle)

- **PostgreSQL** (libre, licencia BSD)

- **SQLite** (libre)

Sistema de Gestión de Base de Datos Relacionales



Orígenes y Evolución de SQL

- En 1970 Edgar F. Cood propone el modelo relacional y asociado a este un sublenguaje de acceso a los datos basado en el cálculo de predicados.
- En los laboratorios de IBM, entre 1974 y 1975 se implementó en un prototipo llamado SEQUEL-XRM.
- Entre 1976 y 1977, condujeron a una revisión del lenguaje (SEQUEL/2)
- Cambió de nombre por motivos legales. convirtiéndose en SQL.
- A partir de 1981, IBM comenzó a entregar sus productos relacionales. En 1983 empezó a vender DB2
- **En 1986, el ANSI adoptó SQL como estándar para los lenguajes relacionales.**
- En 1987 se transformó en estándar ISO, con el nombre de SQL/86.
- Luego se presento la versión SQL/89.
- En 1992 se lanza un nuevo estándar ampliado y revisado del SQL llamado "SQL-92" o "SQL2". El soporte al estándar SQL-92 es general y muy amplio.
- En 1999 se lanza el estándar SQL:1999 con el alias SQL3.
- SQL:2008 y SQL:2012 son las ultimas revisiones al estándar SQL

Lenguaje SQL, grupos de sentencias

DML (*Data Manipulation Language*)

Lenguaje de Manipulación de Datos.

DDL (*Data Definition Language*)

Lenguaje de Definición de Datos.

DCL (*Data Control Language*)

Lenguaje de Control de Datos.

DML

- El **Lenguaje de Manipulación de Datos** (*Data Manipulation Language*, o *DML* en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

- Sentencias: Insert; Delete; Update; Select

DDL

- El **Lenguaje de Definición de Datos** (*Data Definition Language*, o *DDL* en inglés), es el que se encarga de la modificación de la estructura de los objetos de la base de datos.

- Sentencias: Create, Alter, Drop y Truncate.

DCL

- El **Lenguaje de Control de Datos** (*Data Control Language* o DCL en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la base de datos.

- Sentencias: Grant, Revoke y Commit, Rollback

Lenguaje SQL

El Lenguaje SQL permite:

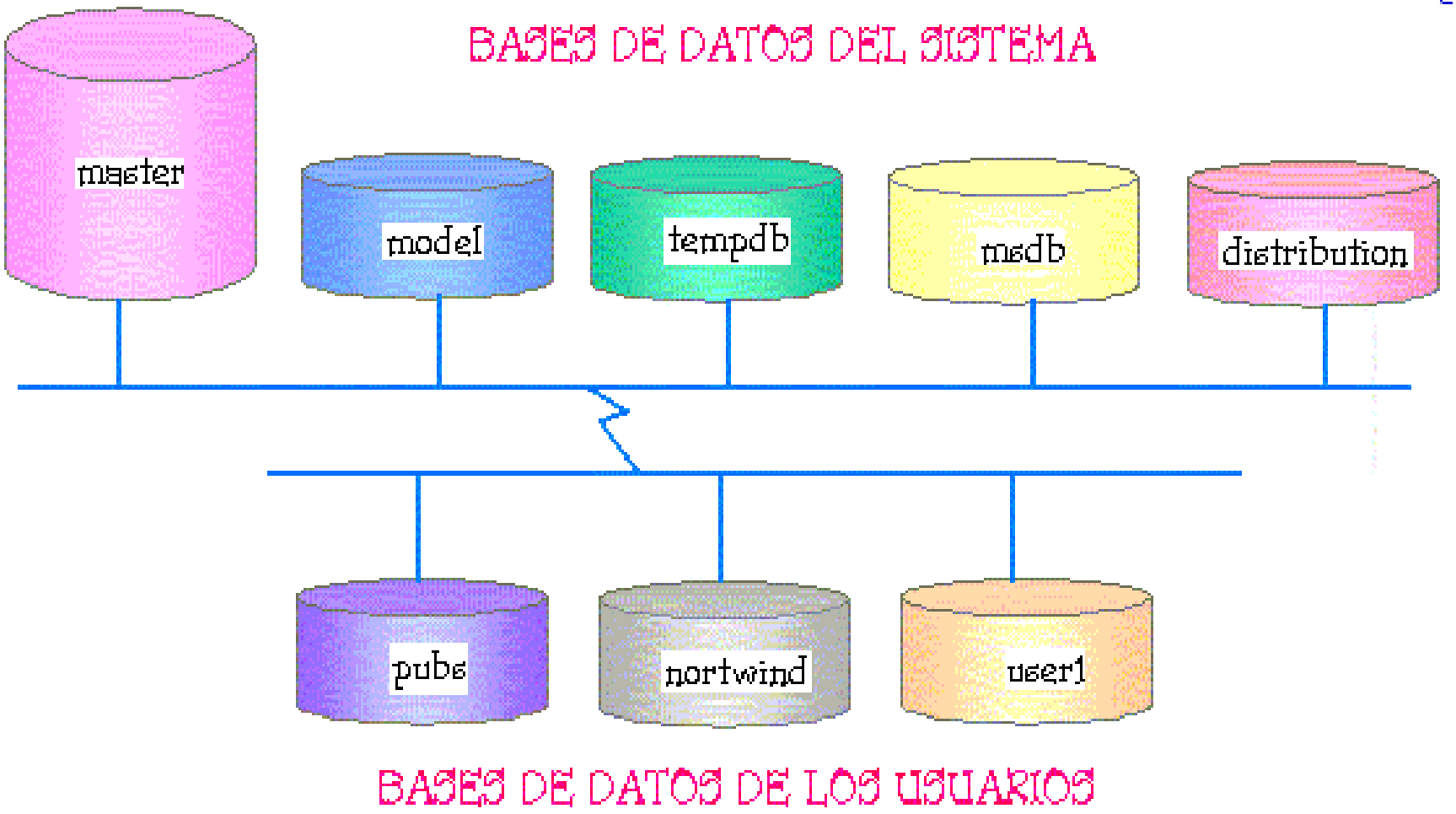
- Definir y destruir objetos de la Base de Datos
- Conceder y denegar autorizaciones para usar estos objetos.
- Consultar y actualizar datos.

Ejemplo: Tabla Pedidos

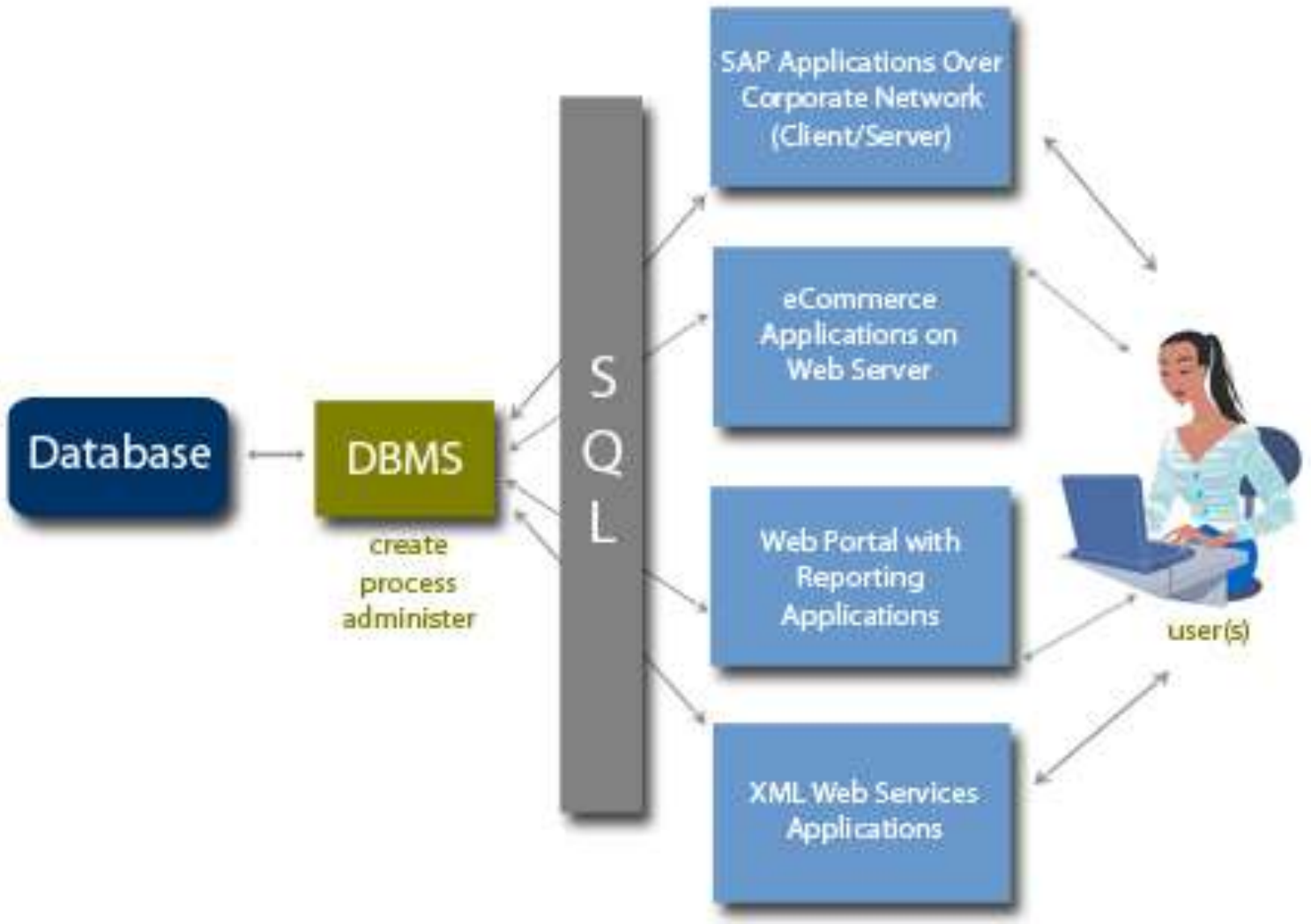
IdPedido	IdCliente	IdEmpleado	FechaPedido	FechaEntrega	FechaEnvío	FormaEnvío	Cargo	Destinatario	Dire
10248	WILMK	5	04/07/1996	01/08/1996	16/07/1996	3	32,38	Wilman Kala	Kes
10249	TOMSP	6	05/07/1996	16/08/1996	10/07/1996	1	11,61	Toms Spezialitäten	Luis
10250	HANAR	4	08/07/1996	05/08/1996	12/07/1996	2	65,83	Hanari Carnes	Rua
10251	VICTE	3	08/07/1996	05/08/1996	15/07/1996	1	41,34	Victuailles en stock	2, r
10252	SUPRD	4	09/07/1996	06/08/1996	11/07/1996	2	51,3	Suprêmes délices	Bou
10253	HANAR	3	10/07/1996	24/07/1996	16/07/1996	2	58,17	Hanari Carnes	Rua
10254	CHOPS	5	11/07/1996	08/08/1996	23/07/1996	2	22,98	Chop-suey Chinese	Hau
10255	RICSU	9	12/07/1996	09/08/1996	15/07/1996	3	148,33	Richter Supermarkt	Sta
10256	WELLI	3	15/07/1996	12/08/1996	17/07/1996	2	13,97	Wellington Importa	Rua
10257	HILAA	4	16/07/1996	13/08/1996	22/07/1996	3	81,91	HILARIÓN-Abastos	Car
10258	ERNSH	1	17/07/1996	14/08/1996	23/07/1996	1	140,51	Ernst Handel	Kirc
10259	CENTC	4	18/07/1996	15/08/1996	25/07/1996	3	3,25	Centro comercial M	Sier
10260	OTTIK	4	19/07/1996	16/08/1996	29/07/1996	1	55,09	Ottilies Käseladen	Mel
10261	QUEDE	4	19/07/1996	16/08/1996	30/07/1996	2	3,05	Que Delícia	Rua
10262	RATTC	8	22/07/1996	19/08/1996	25/07/1996	3	48,29	Rattlesnake Canyo	281
10263	ERNSH	9	23/07/1996	20/08/1996	31/07/1996	3	146,06	Ernst Handel	Kirc
10264	FOLKO	6	24/07/1996	21/08/1996	23/08/1996	3	3,67	Folk och fä HB	Åke

Tablas de una Base de Datos Relacional

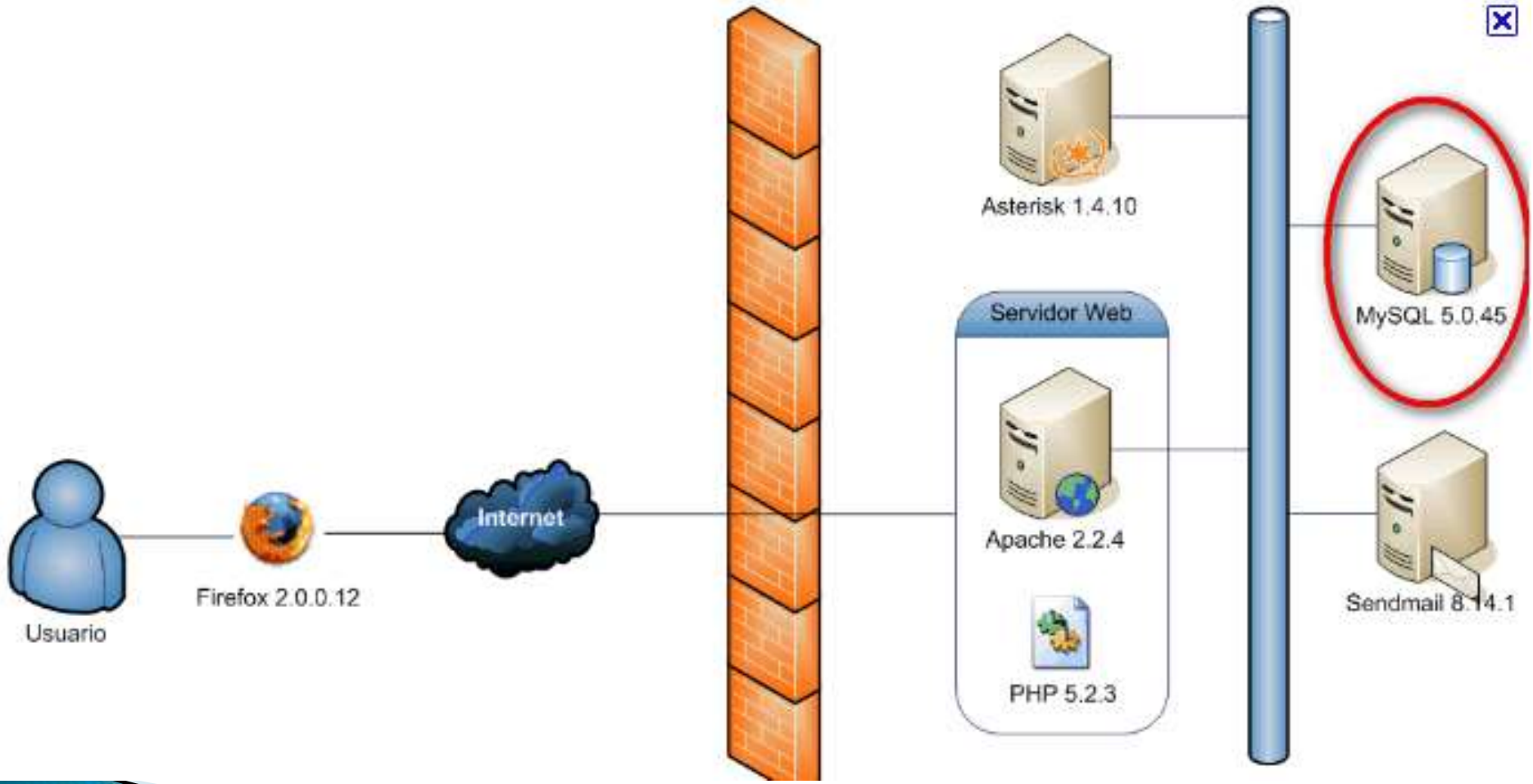
Base de Datos

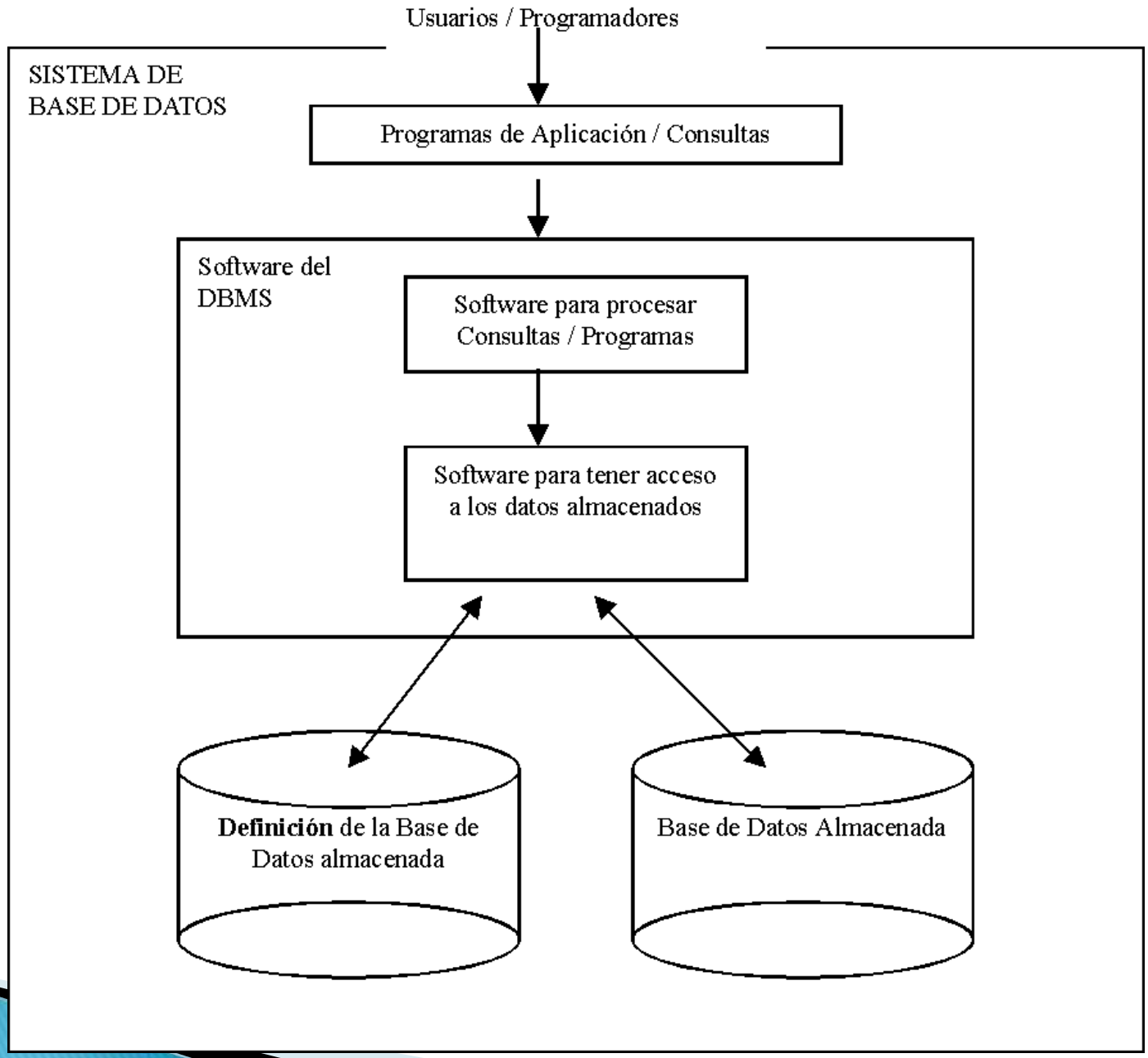


Usos de SQL

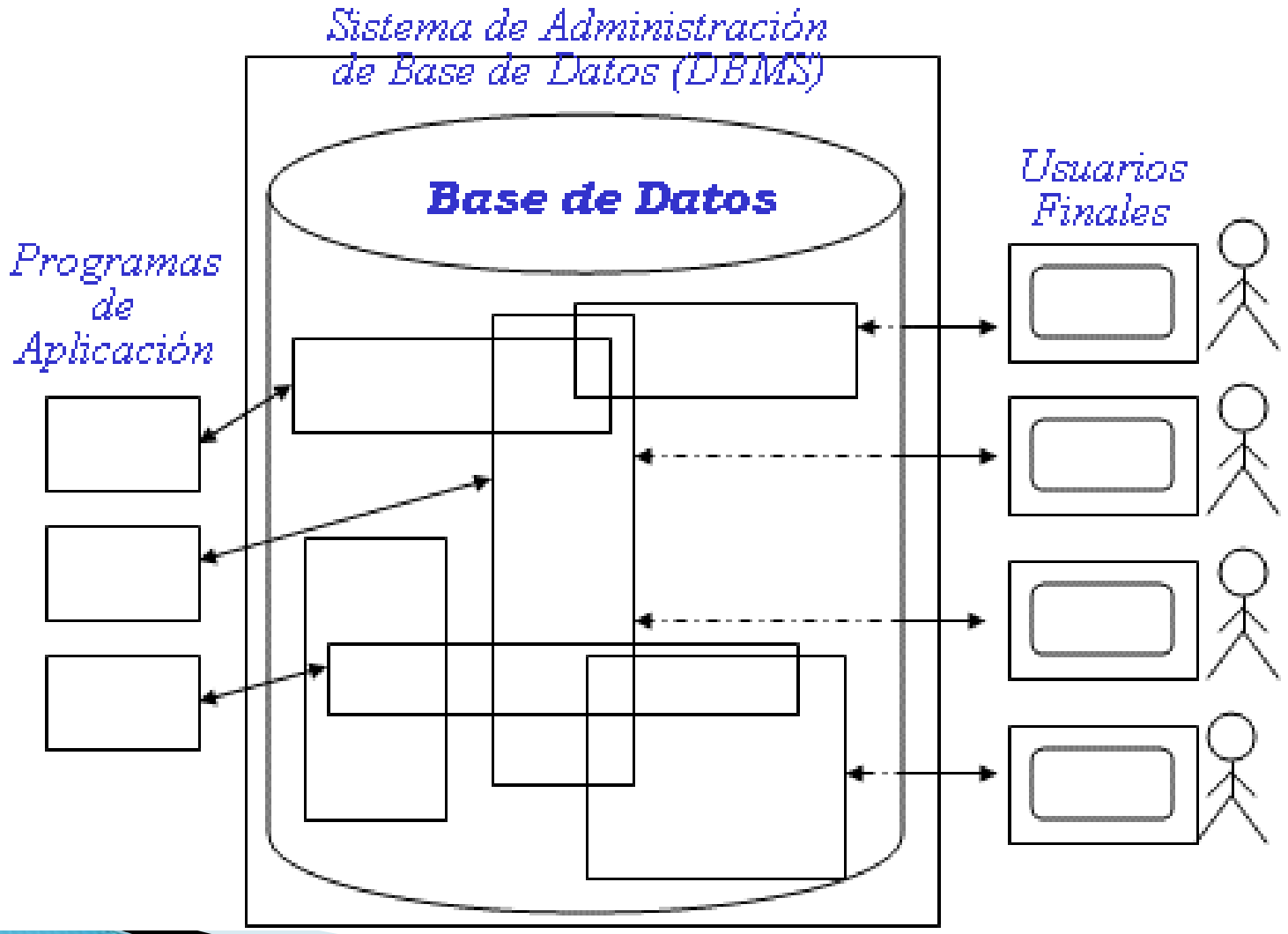

SAP Applications Over Corporate Network (Client/Server)
eCommerce Applications on Web Server
Web Portal with Reporting Applications
XML Web Services Applications
S Q L
Database
DBMS
create
process
administer
user(s)

Ejemplo de uso de un SGBD



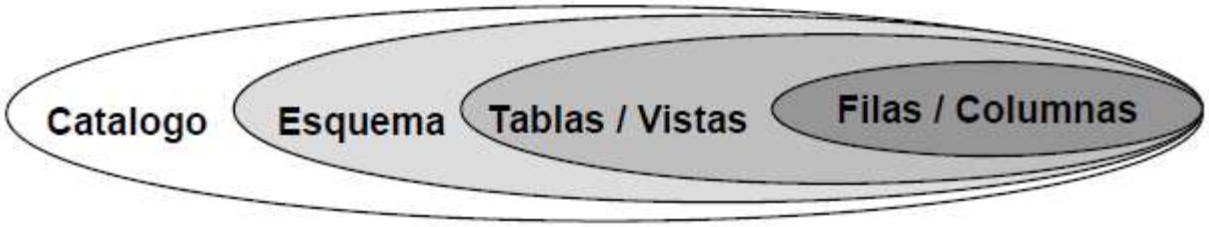

Usuarios / Programadores
SISTEMA DE
BASE DE DATOS
Programas de Aplicación / Consultas
Software del
DBMS
Software para procesar
Consultas / Programas
Software para tener acceso
a los datos almacenados
Definición de la Base de
Datos almacenada
Base de Datos Almacenada

DBMS


Sistema de Administración
de Base de Datos (DBMS)
Base de Datos
Programas
de
Aplicación
Usuarios
Finales

Base de Datos, Modelo catálogo/esquema

- En SQL las bases de datos se organizan mediante una jerarquía de contenedores.



- ANSI SQL define un modelo de catálogo/esquema para almacenes de datos.
- **Catálogo**: Un catálogo es otra forma de denominar una base de datos. Es una colección de esquemas relacionados.
- **Esquema**: Un esquema es una colección de objetos de base de datos de los que es propietario o autor un usuario particular.
- **Tabla** : Las tablas son colecciones de columnas dispuestas en un orden específico.

Base de Datos, Modelo catálogo/esquema

- Un Catálogo contiene un conjunto de Esquemas.
- Un Esquema contiene un conjunto de Tablas.
- Los nombres dentro de un contenedor deben ser únicos, pero otros contenedores pueden contener objetos con el mismo nombre. Así por ejemplo, dos tablas pueden tener un atributo con el mismo nombre.

Base de Datos, Modelo catálogo/esquema

- Usando la jerarquía, se puede dar un nombre totalmente **calificado** a un objeto de la base de datos de la siguiente manera:

<catalogo>.<esquema>.<tabla>.<columna>

- Aun cuando no es parte del estándar, la mayoría de los SGBD define un contenedor denominado base de datos para contener tablas o esquemas que contienen tablas.

SQL Componentes de una Sentencia

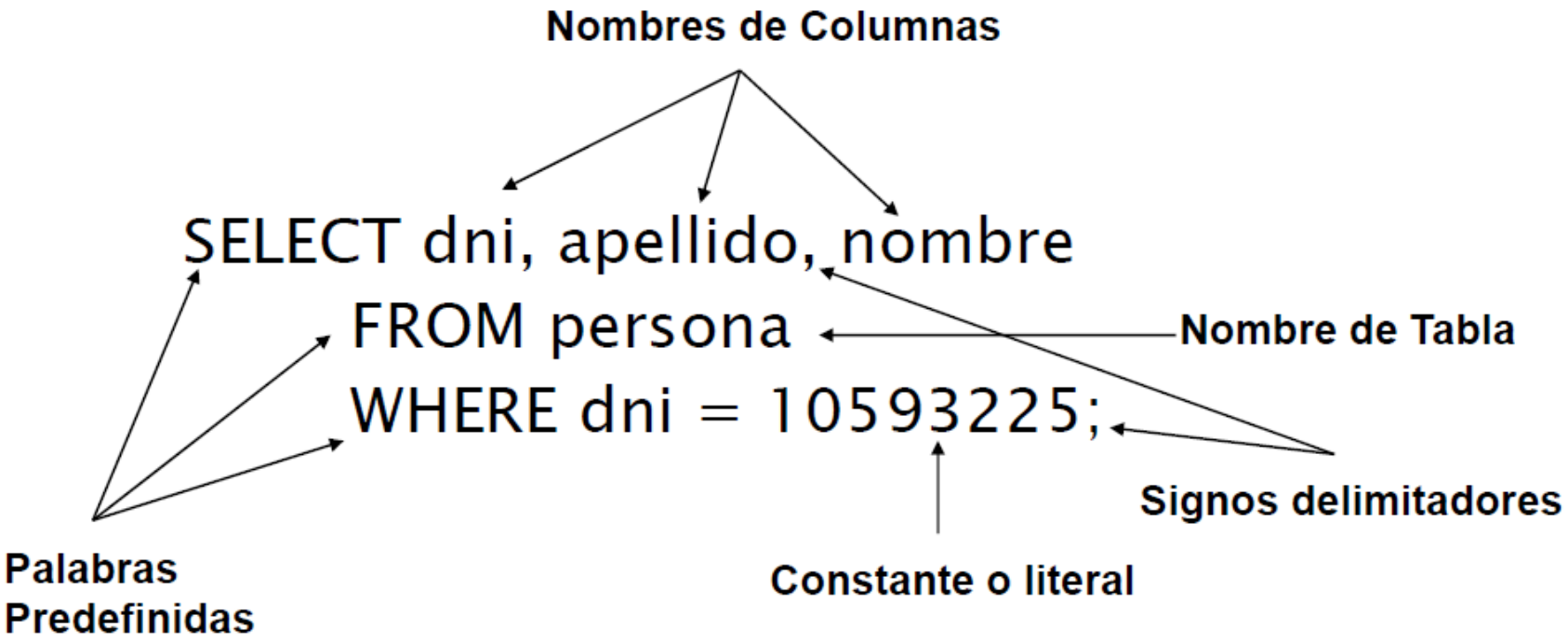
Palabras reservadas: son las que tienen un significado propio en el lenguaje. Todas las sentencias SQL comienzan por una palabra reservada. Ejemplo de palabras predefinidas serian SELECT, INSERT, WHERE, INTO.

Palabras definidas por el usuario: son los nombres de tablas y columnas

Constantes o literales: representan un valor determinado.

Signos delimitadores: son signos especiales que aparecen en las sentencias fuera de los otros elementos. Sirven para separar o delimitar estos últimos. Entre ellos se encuentran los espacios en blanco, los paréntesis, las comas y otros.

SQL Componentes de una Sentencia – Ejemplo



SQL-Consideraciones generales sobre las Sentencias (I)

Los espacios en blanco no son significativos: para aumentar la legibilidad podemos separar las sentencias en distintas líneas, una para cada parte de la sentencia. Para ello podemos usar tabuladores, espacios en blanco.

Las sentencias SQL no distinguen entre mayúsculas y minúsculas: para aumentar la legibilidad utilizaremos **mayúsculas para las palabras claves** de SQL y minúsculas para los nombres de las tablas y las columnas. Sin embargo, SQL no hace distinciones para las palabras claves como así tampoco para los nombres de las columnas ni de las tablas.

SQL-Consideraciones generales sobre las Sentencias (II)

Los puntos y coma son opcionales: las sentencias SQL pueden ser terminadas por un punto y coma (";"); siendo utilizadas para separar múltiples sentencias SQL.

SQL acepta dos tipos de comentarios:

- Los **comentarios de línea**, los cuales comienzan con dos signos menos (--)
- Los **comentarios multilíneas**, los cuales comienzan con /* y finalizan con */

SQL Sentencias y Clausulas

Todas las sentencias SQL comienzan con un verbo, una palabra clave que describe lo que la sentencia hace. CREATE, INSERT, DELETE, UPDATE son verbos típicos. La sentencia continua con una o más cláusulas. Una cláusula puede especificar los datos sobre los que debe actuar la sentencia, o proporcionar más detalles acerca de lo que la sentencia debe hacer.

Todas las cláusulas comienzan también con una palabra clave, tal como WHERE, FROM, INTO y HAVING. Algunas cláusulas son opcionales, otras obligatorias. La estructura y contenido específico varían de una cláusula a otra. Muchas cláusulas contienen nombres de tablas o columnas; algunas pueden contener palabras claves adicionales, constantes o expresiones

SQL – 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** **OR** **NOT**		Los operadores lógicos permiten comparar expresiones lógicas devolviendo siempre un valor verdadero o falso.Los operadores lógicos se evaluan de izquierda a derecha.
Concatenación	+		Se emplea para unir datos de tipo alfanúmerico.

Precedencia de Operadores

Tipo	Operador	Símbolo
Grupo	Agrupación principal	( )
Aritmético	Multiplicativo	* / %
Aritmético	Aditivo	- +
Otros	Concatenación de cadenas	+
Lógico	NOT	NOT
Lógico	AND	AND
Lógico	OR	OR

Precedencia de Operadores (MySql)

```
:=
||, OR, XOR
&&, AND
NOT
BETWEEN, CASE, WHEN, THEN, ELSE
=, <=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP, IN
|
&
<<, >>
-, +
*, /, DIV, %, MOD
^
- (resta unaria), ~ (inversión de bit unaria)
!
BINARY, COLLATE
```

La lista muestra de Menor a Mayor precedencia.

SQL – Tipos de datos numéricos

ANSI SQL	MySQL	Rango de valores
SMALLINT	SMALLINT[(M)]	-32768 a 32767
INT INTEGER	INT[(M)] INTEGER[(M)]	-2147483648 a 2147483647
BIGINT	BIGINT[(M)]	-9223372036854775808 a 9223372036854775807

(M) indica el ancho máximo para mostrar. El valor máximo es 255.

SQL – Tipos de datos numéricos

ANSI SQL	Rango Valores	MySQL	SQL Server
DECIMAL[(P [, S])]	-10^38+1 hasta 10^38-1	decimal	decimal
NUMERIC[(P [, S])]	-10^38+1 hasta 10^38-1	numeric	decimal

SQL – Tipos de datos numéricos

ANSI SQL	MySQL	Rango de valores
FLOAT(P)	FLOAT(P)	número en punto flotante con signo con una precisión de al menos P dígitos.
DOUBLE PRECISION	DOUBLE	numero en punto flotante con signo de precisión mayor que real
REAL	REAL	numero en punto flotante consigno de precisión simple

SQL – Tipos de datos carácter

ANSI SQL	Descripción	MySQL	SQL Server
Character [(n)]	Se puede especificar la cantidad de caracteres que se desean almacenar. Si se almacenan menos de los permitidos, el RDBMs le agrega espacios para completarlo.	char	char
Character Varying [(n)]	Igual que el character excepto que la longitud es variable. Solo utiliza memoria para la cantidad de caracteres almacenados	varchar	varchar

SQL – Tipos de datos temporales

ANSI SQL	Descripción	MySQL	SQL Server
Date	Permite almacenar una fecha como año, mes y día	date, datetime	datetime, smalldatetime
Time	Permite almacenar una hora como hora, minuto, segundos	time	datetime
Timestamp	Permite almacenar fecha y hora como año, mes, día, hora, minutos y segundos	timestamp	

SQL – Otros tipos de datos

ANSI SQL	MySQL	Rango de valores
BIT	BIT	Almacena un numero fijo de números binarios.
BIT VARYING(L)		L cantidad de bit
BINARY LARGE OBJECT[(L)] o BLOB	BLOB	Datos binarios hasta un max. de 65536 byte. (ejem. imágenes, sonido)
TEXT	TEXT[(M)]	Texto hasta 65536 caracteres
BOOLEAN BOOL	BOOLEAN TINYINT(1)	Verdadero / Falso

DDL – Lenguaje de definición de datos

Comandos:

CREATE: Permite crear una Base de Datos, Tabla, Indice.

DROP: Permite borrar una Base de Datos, Tabla, Indice.

ALTER: Permite modificar la estructura de una Base de Datos, Tabla, Indice.

RENAME: Permite cambiar el nombre a una tabla.

DDL – Comando para Crear Base de Datos

CREATE DATABASE <nombre_base_de_datos>

Ejemplo:
CREATE DATABASE Alumnos;

DDL – Comando para Borrar Base de Datos

DROP DATABASE
  <nombre_base_de_datos>

Ejemplo:
DROP DATABASE Alumnos;

DDL – Comando para Crear una Tabla en la Base de Datos

```
CREATE TABLE
<nombre_de_la_tabla>
 ( <nombre_columna-1> < tipo>,
  <nombre_columna-2> < tipo>,
  … ,
  <nombre_columna-n>  <tipo> );
```

DDL – Comando para Crear una Tabla en la Base de Datos

Ejemplo:

```
CREATE TABLE  Carrera
(  CarrCod  Int,
CarrDescripcion VARCHAR(40)  );
```

DDL – Comando para Borrar una Tabla en la Base de Datos

DROP TABLE <nombre_de_la_tabla>

Ejemplo:
DROP TABLE Carrera;

SQL – Comandos varios (MySql)

USE <nombre_base_de_datos>

Permite seleccionar la base de datos con la que se va a trabajar.

Ejemplo: USE Alumnos;

SHOW DATABASES;

Muestra las bases de datos existentes.

SHOW TABLES;

Muestra las tablas de la bases de datos seleccionada.

SQL – Comandos varios (MySql)

DESCRIBE <nombre_de_tabla>;

Muestra la estructura de la tabla, o sea el nombre y el tipo de cada uno de los campos.

Ejemplo: DESCRIBE carrera;

SQL – INSERT –Comando para agregar datos a una Tabla

INSERT INTO <nombre_tabla>
[(< atributo1>, . . . , <atributoN>)]
VALUES (<expresion1>, ..,<expresionN>);

Ejemplo:

- INSERT INTO alumno
  (aluDoc, aluNom, aluDir)
  VALUES (30111222, 'Lopez Martina', 'Av. Roca 77');

- INSERT INTO carrera VALUE (10, "PUI");

DML – SELECT–Comando para consultar datos de una Tabla

SELECT <lista_atributos> FROM <tablas>
  WHERE <condicion> ;

Ejemplo:

```
SELECT  *  FROM alumno
WHERE  aluDoc  = 32333444;

SELECT aluDir, aluNom FROM alumno
WHERE aluDoc > 30111222;
```

DML – SELECT

SELECT [DISTINCT | ALL] <lista columnas>
FROM <lista tablas>
[WHERE<condición de selección de filas>]
[ORDER BY <especificación de ordenamiento> [ASC | DESC]] ;

SELECT: Permite recuperar información de una o mas tablas de la Base de Datos.

Devuelve una **tabla resultado** con las columnas indicadas en **<lista columnas>** desde las tablas indicadas en **FROM <lista tablas>** que cumplan con la condición **WHERE<condición de selección de filas>** ordenadas por las columnas indicadas en **ORDER BY <especificación de ordenamiento>**

DML – SELECT, Clausulas

DISTINCT: Elimina las filas duplicadas del resultado de la consulta.

<lista columnas> lista de atributos que se desea recuperar. El orden en que se especifican los atributos determina el orden de los atributos en los resultados.

El **"*"** (asterisco) es un comodín que indica todas las columnas.

FROM <lista tablas> : Nombre de la o las tablas sobre las que se desea realizar la consulta.

Ejemplo:

SELECT * FROM alumno WHERE aluNota > 3;

DML – SELECT, Clausulas

[WHERE<condición de selección de filas>] : Condiciones que deben cumplir las tuplas para formar parte de la selección.

Estructura de la cláusula WHERE:

WHERE <columna> <operador> <valor>

Operador	Significado	Uso
=	Igual	Apellido = 'PEREZ'
> , <	Mayor que, Menor que	Sueldo > 3000
>= , <=	Mayor o igual que, Menor o igual que	Sueldo <= 1500
<>	Distinto	Documento<> 17505822

DML – SELECT, Clausulas

[ORDER BY <especificación de ordenamiento> [ASC | DESC]] : Se indican los atributos por los cuales se desea obtener ordenada la consulta.

DML – SELECT, Ejemplo

/*Recupera el documento, nombre y dirección de todos los empadronados que tengan documento > 20000000 , ordenado por nombre en forma descendente */

```
SELECT documento, nombre, direccion
  FROM padron
  WHERE documento > 20000000
  ORDER BY nombre DESC;
```

DML – SELECT, Ejemplo

/*Recupera el nombre de todos los empadronados que tengan documento > 20000000 , ordenado por nombre, pero no muestra los repetido */

```
SELECT DISTINCT nombre
  FROM padron
  WHERE documento > 20000000
  ORDER BY nombre;
```

DML – SELECT, Operador Like

SQL permite la búsqueda de patrones de información en la base de datos, por ejemplo si queremos conocer todos los nombre que terminan en “rio” o aquellos que comienzan con “Ma”, no lo podríamos hacer utilizado condiciones como Where nombre = ‘rio’, porque solo obtendríamos como resultado aquellos que se llaman ‘rio’.

Estas consultas se pueden realizar utilizando en la condición el operador **[NOT] LIKE** y los **comodines “%” y “_”** .

```
SELECT documento, nombre
  FROM padron
  WHERE nombre Like ‘%rio’
  ORDER BY nombre;
```

DML – SELECT, Ejemplo

Los caracteres comodín son:

Comodín	Descripción
%	Concuerda con cualquier substring conteniendo cero ó mas caracteres
_	(guion bajo) Concuerda con un único carácter

Ejemplo

WHERE nombre LIKE '%rio'

Un select con esta condición devuelve todos los nombres que terminan en 'rio', el comodín '%' reemplaza a cualquier carácter que posea 'nombre' al comienzo.

DML – SELECT, Ejemplo

```
/*Recupera el nombre de todos los empadronados cuyo nombre comience
  con 'G', ordenado por nombre, pero no muestra los repetido */

SELECT DISTINCT nombre
  FROM padron
  WHERE nombre Like 'G%'
  ORDER BY nombre;

/*Recupera el documento y nombre de todos los empadronados
  cuyo nombre comience con 'Juli_', ordenado por documento.
  */

SELECT documento, nombre
  FROM padron
  WHERE nombre Like 'Juli_'
  ORDER BY documento;
```

Preguntas ?