



Parte 1

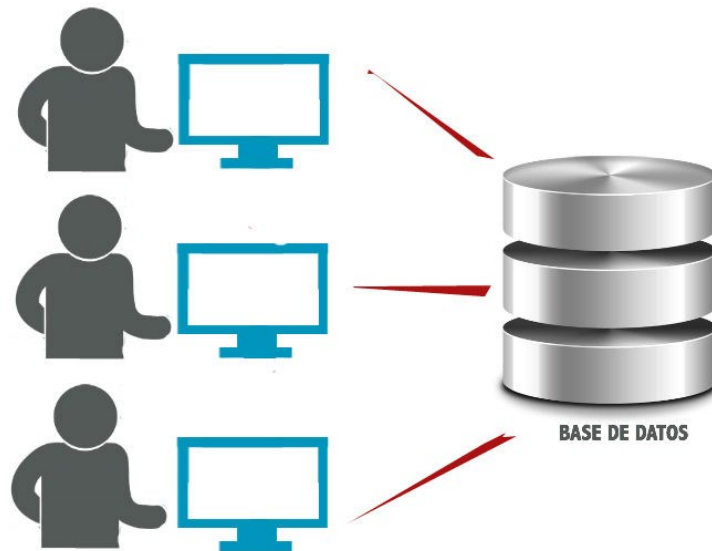
Bases de Datos SQL Server



¿Que es una base de datos?

Base de Datos (BD)

Es una colección de datos relacionados y estructurados entre si; la recopilación de la información en forma ordenada dentro de una BD, nos ayuda a realizar funciones de acceso y búsqueda de los datos de una manera fácil y sencilla.



Características de una BD

Seguridad y
Privacidad

Mínima
Redundancia

Disponibilidad
y
Accesibilidad

Tiempo de
Respuesta
Rápido



Base de datos

Acceder

Administrar

		Columnas			
		0	1	2	3
Filas	0	1	3	5	7
	1	5	4	1	16
	2	7	9	61	13
	matrizDeEnteros [3][4]				

Modificar

Actualizar

Controlar y

Organizar fácilmente los datos

Tabla EmployeeMaster

EmployeeID	FirstName	AddressID	ShiftID	LastName	MiddleName	SSN	...
1	Sheri	1	1	Nowmer	E	245797967	...
2	Derrick	2	1	Whelply	R	509647174	...
3	Michael	3	1	Spence	C	42487730	...
4	Maya	4	1	Gutierrez	Y	56920285	...
5	Roberta	5	1	Damstra	B	695256908	...

Ver

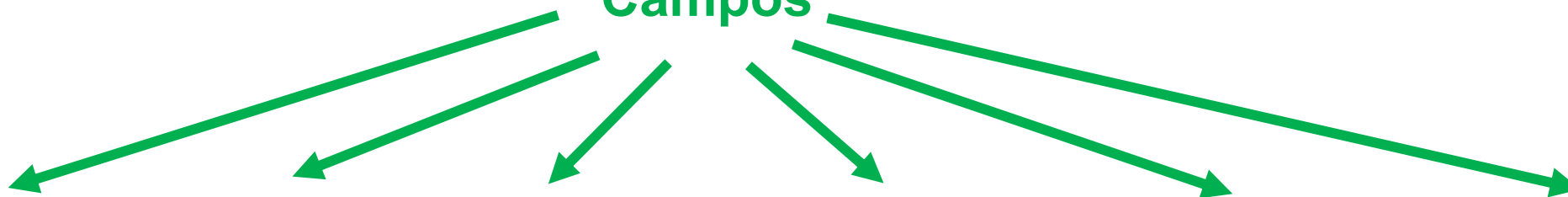
FirstName	LastName	Description
Sheri	Nowmer	Engineering
Derrick	Whelply	Engineering
Michael	Spence	Engineering

Tabla Department

DepartmentID	Description	rowguid
1	Engineering	3FFD2603-EB6E-43B2-A8EF-C4F5C3064026
2	Tool Design	AE948718-D4BF-40E0-8ECD-2D9F4A0B211E
3	Sales	702C0EE3-03E6-4F95-9AB8-99F4F25921F3
4	Marketing	3E3C4476-B9EC-43CB-AA12-1E7A140A71A4
5	Purchasing	D6C63691-93B5-4F43-AD88-34B6B9A3C4A3

Elementos Básicos en una BD

Campos

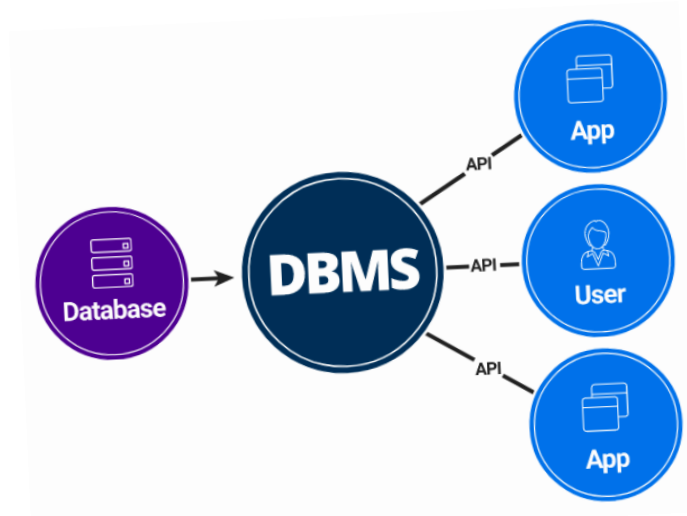


Registros

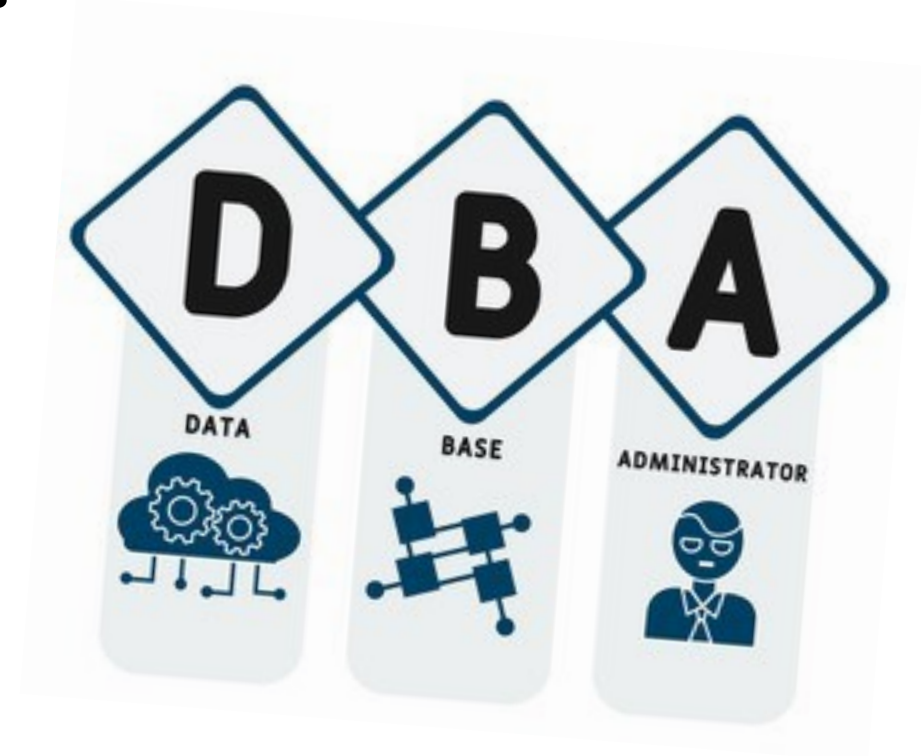
	id	nombre	primerApellido	segundoApellido	correo	telefono	fechaNacimiento	curp
↓	1	Marcelo Isai a	García	mirón	marcelo@outlook.com	9911362600	1997-12-12	MADA971212HVZRMN04
↓	2	Oliver Alexis	Martínez	Estudillo	alexis@gmail.com	8897877417	1996-04-18	DIAE960418HOC SVL07
↓	3	Oscar	Mendoza	garcía	omscar@outlook.es	7711589568	1994-04-07	RUVJ940407HOCSSN03
	4	Edgar	Martínez	Espinoza	edgargmail.com	5528356144	1996-05-23	DOML960323HMNMTS00
	5	Rodrigo	Tolentino	Martínez	rodrigo@gmail.com	4421436224	1998-03-13	TOMR980313HHGLRD06
	6	Jesiel	García	Pérez	jesiel@gmail.com	3317901341	1990-11-08	GAPJ901108HHGRRS00
	7	Christian Josue	Gonzalez	Lozano	christian@gmail.com	4922153353	1996-06-19	GOLC960619HZSNZH08
	8	Luis Enrique	Lopez	Cruz	luis@gmail.com	2235700644	1997-07-15	LOCL970715HGTPRS04
	9	Rolando	Marquez	Hernandez	rolando@gmail.com	1168329969	1997-03-08	MAHR97030815HRL600
	10	Jesús Yotecatl	Miranda	Espinosa	jesus@gmail.com	2213335247	1997-06-14	MIEJ970614HMCRSS05

Datos

En una BD intervienen...



Sistema Gestor de la BD
Controla los datos, así como el programa que tiene acceso a estos.



Administrador de la BD
Son los usuarios que controlan y utilizan la BD.

DBMS

DataBase Management System

Sistema de administración de bases de datos



ORACLE®



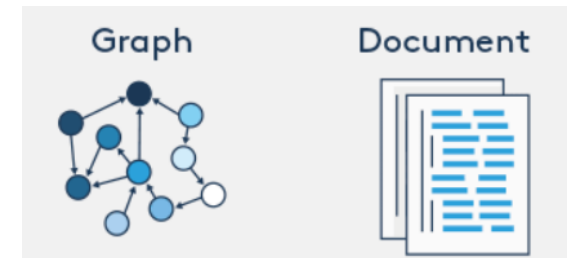
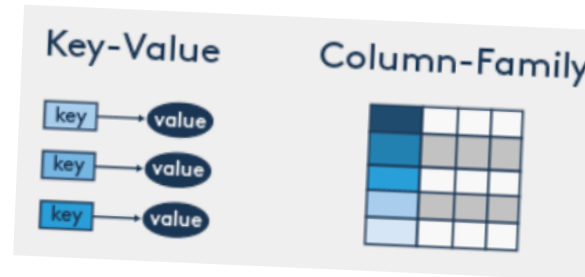
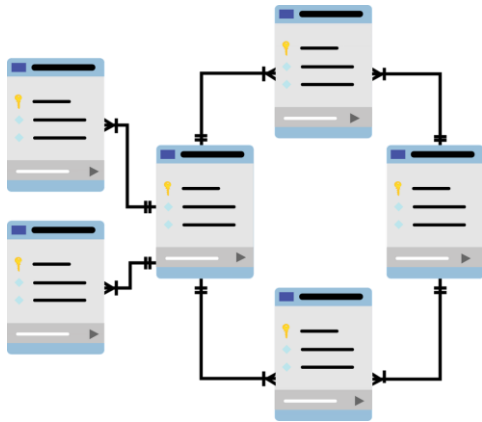
Principales modelos de bases de datos

2

NoSQL y orientadas a objetos

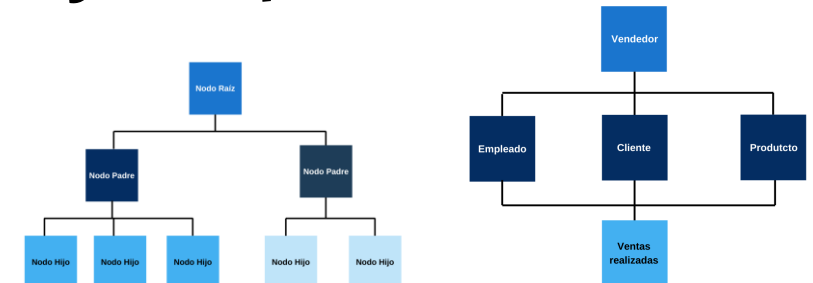
1

Relacional (SQL)



3

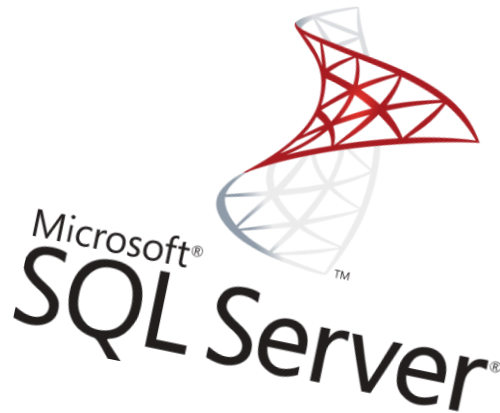
Jerárquico y de Red



1.- Relacional (SQL)



PostgreSQL



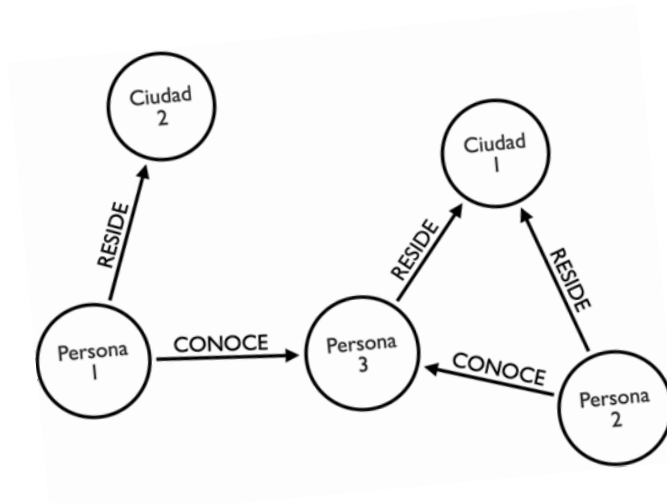
ORACLE

- Generalmente se accede a los datos mediante SQL (Structured query language) es decir, mediante entradas o consultas que “devuelven” un resultado.
- Utilizan sistemas gestores de base de datos como:

MySQL
PostgreSQL

Microsoft SQL Server
Oracle SQL

2.-NoSQL y orientada a objetos



key	value
123	Address@23
126	"Booya"

```

{
  "ID": 1,
  "FIRST": "Frank",
  "LAST": "Weigel",
  "ZIP": "94040",
  "CITY": "MV",
  "STATE": "CA"
}
JSON
    
```

KEY	First	Last	ZIP_id
1	Frank	Weigel	2
2	All	Dodson	2
3	Mark	Azad	2
4	Steve	Yen	3

NoSQL = Not Only SQL = bases de datos que no sólo utilizan SQL

- 🚧 Útiles para procesar datos no estructurados o semiestructurados.
- 🚧 Permiten almacenaje, uso y acceso flexible a los datos.

Ejemplos de bases de datos NoSQL:

- Grafos
- Llave-valor
- Orientada a documentos

2.-NoSQL y orientada a objetos

Base de datos de **grafos**:



- 🚧 Su principal característica es que representan la información en vértices y aristas.
- 🚧 Un grafo está compuesto por dos elementos: nodos (vértices) y relaciones (aristas).
- 🚧 Los nodos representan entidades, allí almacenaremos atributos de tipo clave-valor, y las relaciones representan cómo se conectan y se asocian dos nodos.

Ejemplos:



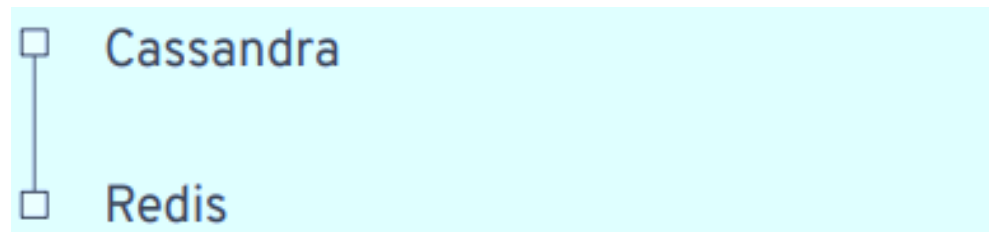
2.-NoSQL y orientada a objetos

Base de datos **llave-valor**:



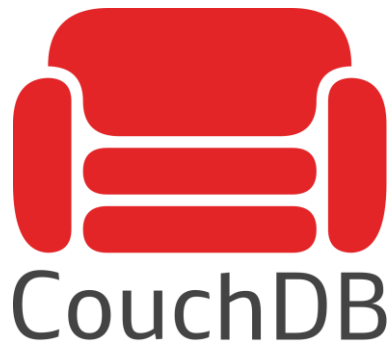
- Utiliza un método de clave o llave-valor para almacenar datos. Las llaves o clave representan una característica identificativa única.
- Gozan del aprecio de desarrolladores y programadores por ser muy efectivas en la consulta y fáciles de escalar.

Ejemplos:



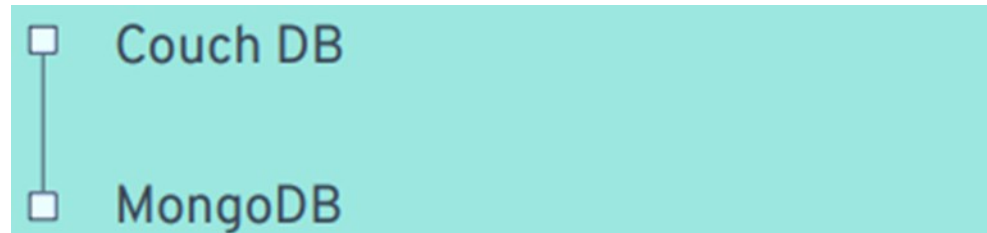
2.-NoSQL y orientada a objetos

Base de datos **orientada a documentos:**

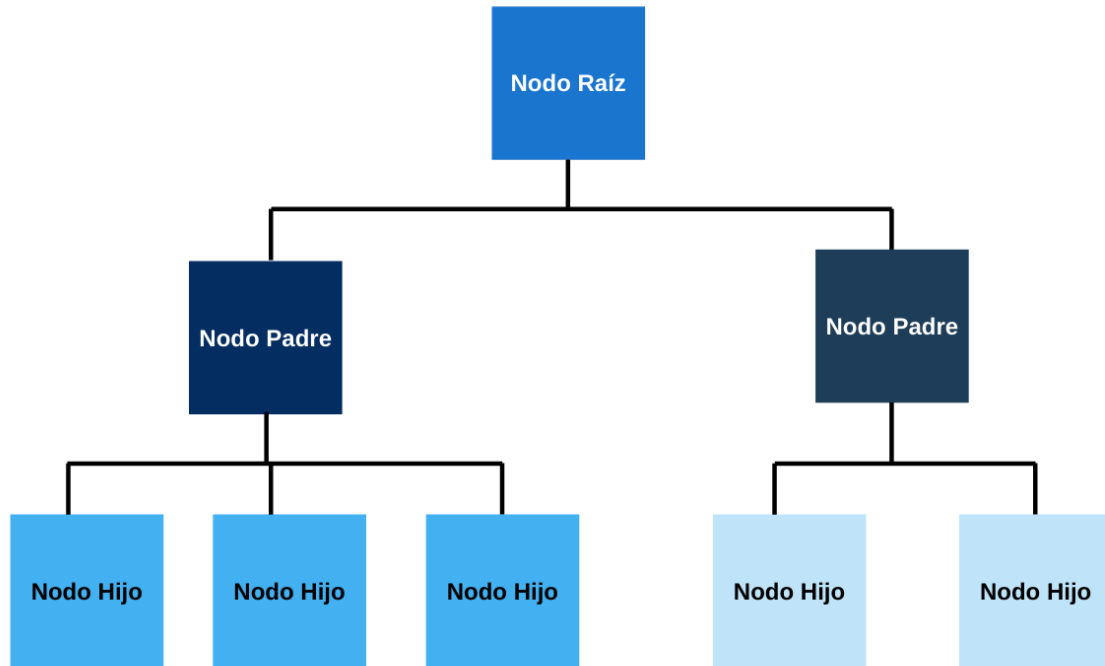


- Se usan para almacenar y consultar datos. Los documentos puede ser de texto, archivos XML o JSON.
- Tiene una naturaleza flexible, semiestructurada y jerárquica de los documentos.
- Funciona bien con casos de uso como catálogos, perfiles de usuario y sistemas de administración de contenido.

Ejemplos:



3.-Jerárquico y de Red

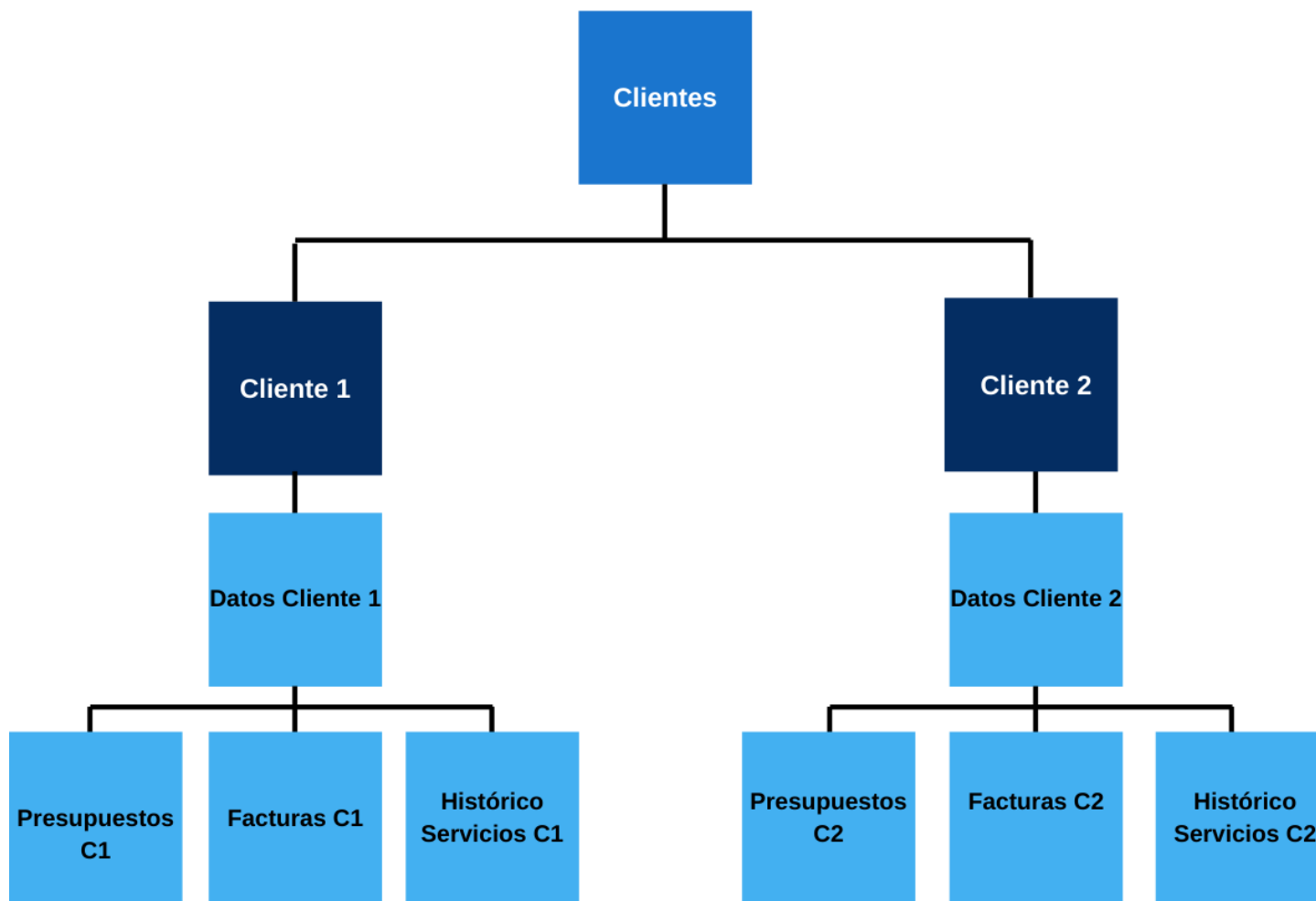


📡 Representa objetos y sus relaciones, donde las dependencias responden a una jerarquía.

📡 Raíz: origen de los datos

📡 Hay tipos de objetos “padres” que tienen “hijos”.

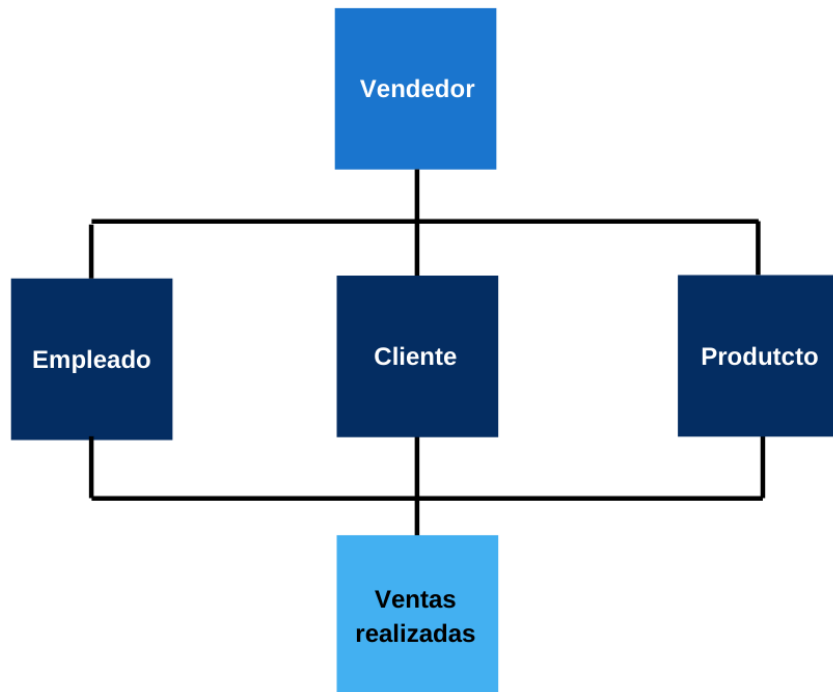
📡 Se ve como un árbol.



Por ejemplo, una empresa puede crear una base de datos jerárquica para organizar a sus clientes. Así, el nodo raíz sería Clientes, cada cliente sería un nodo padre, que su vez podrían lugar a nodos hijos como servicios, facturas, presupuestos, datos, etc.

3.-Jerárquico y de Red

📡 Representa objetos y sus relaciones, sin ningún tipo de jerarquía.

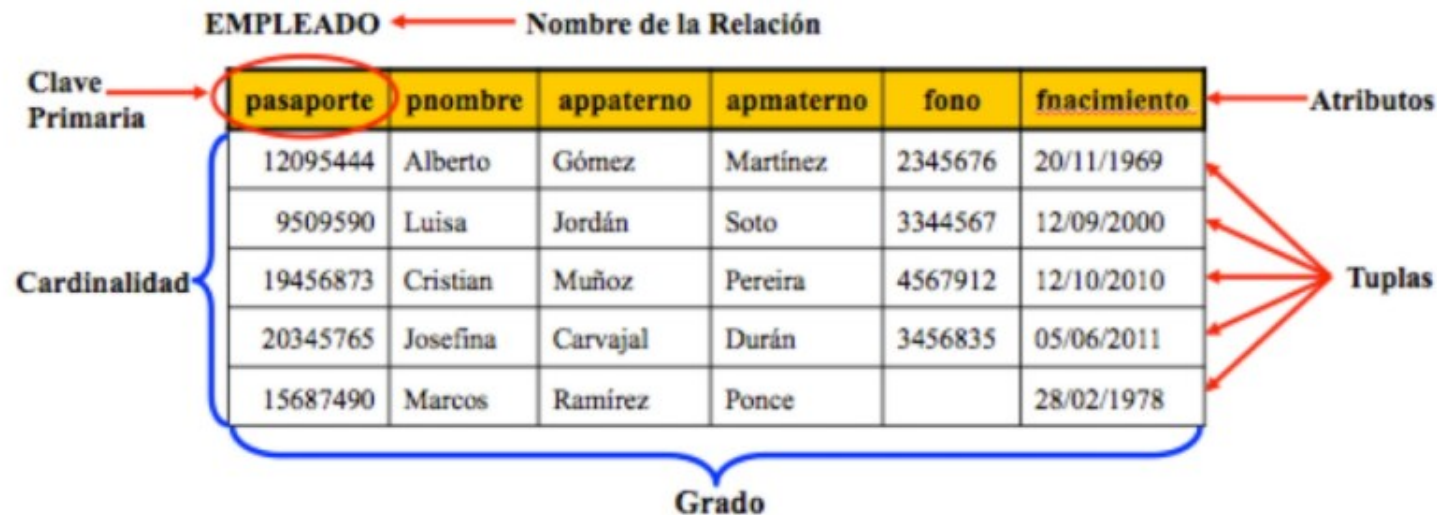


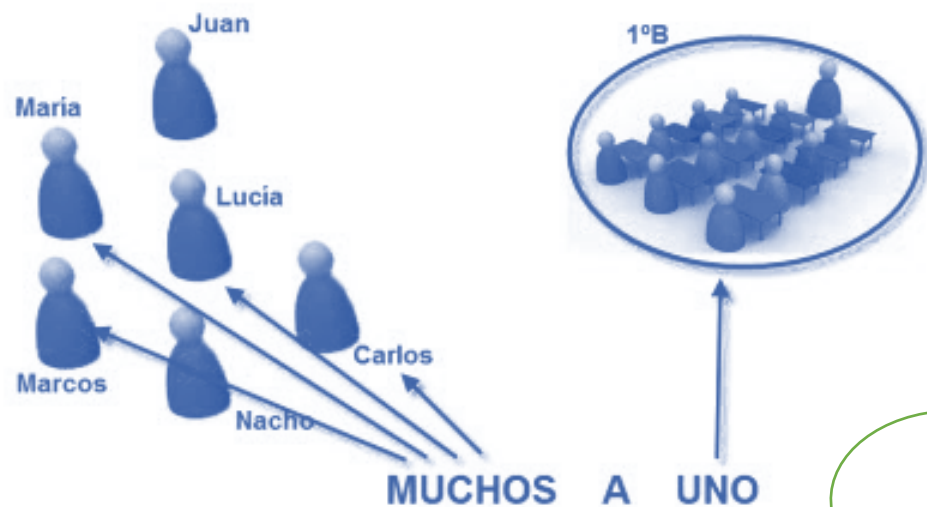
Su esquema consiste en:

- Tipos de objetos: nodos
- Relaciones: arcos

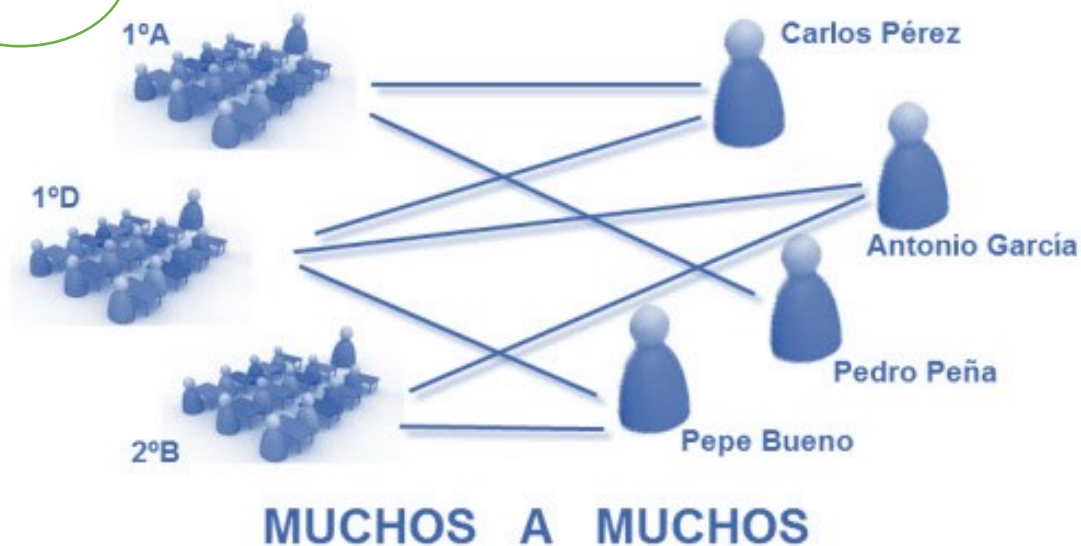
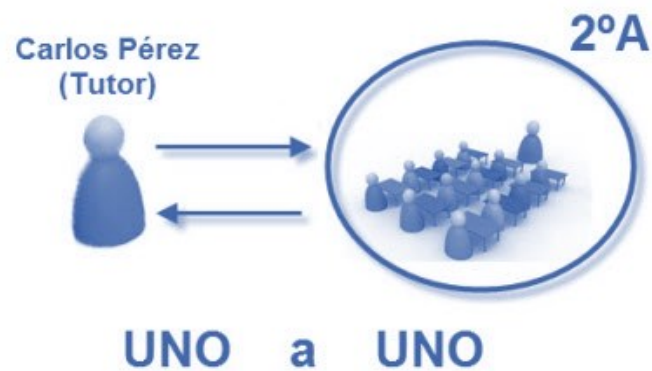
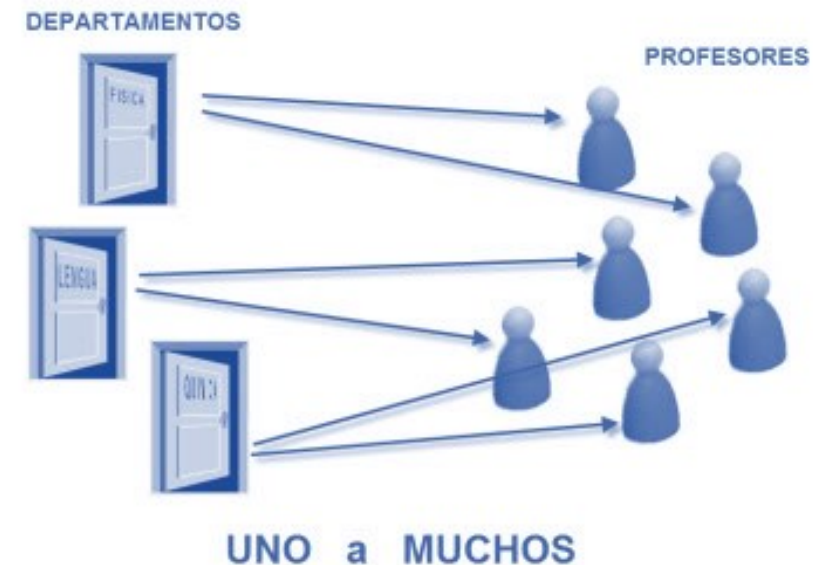
Conceptos dentro del modelo relacional

Terminología Relacional		Terminología de Tablas		Terminología de Archivo
Relación	=	Tabla	=	Archivo
Tupla	=	Fila	=	Registro
Atributo	=	Columna	=	Campo
Grado	=	Número de columnas	=	Número de campos
Cardinalidad	=	Número de filas	=	Número de registros



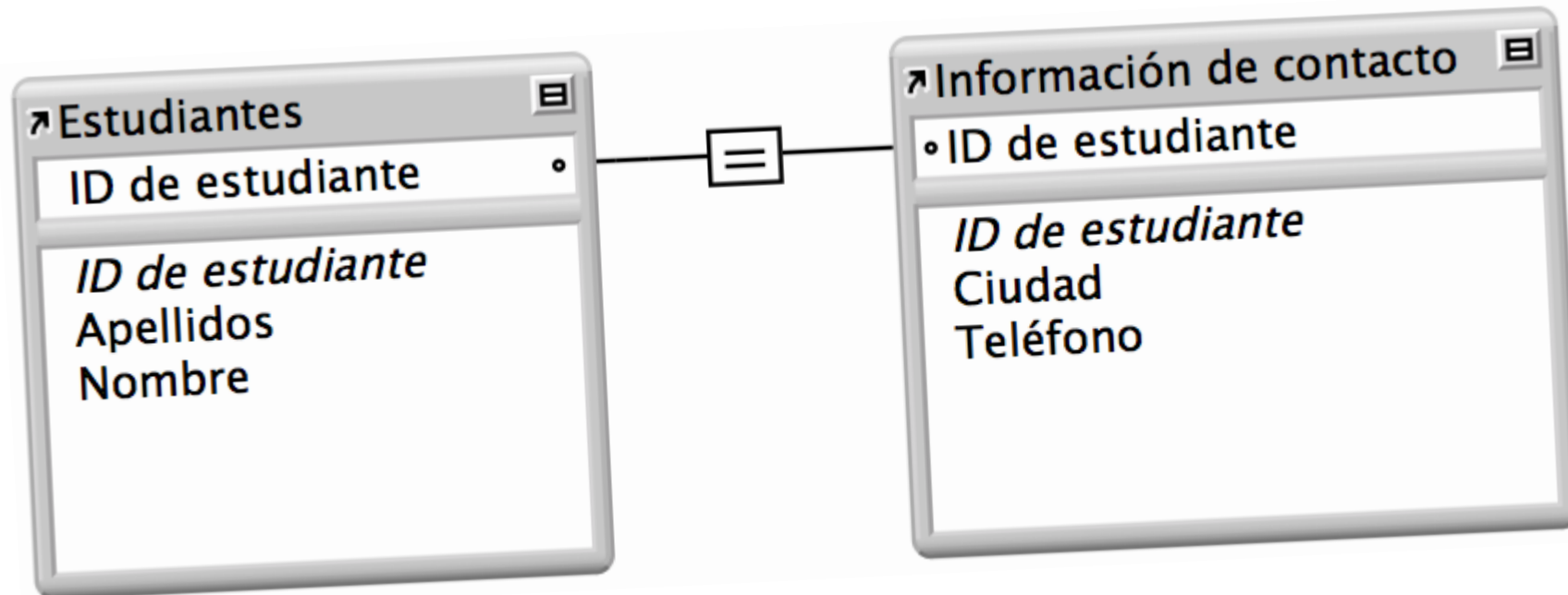


Tipos de Relaciones



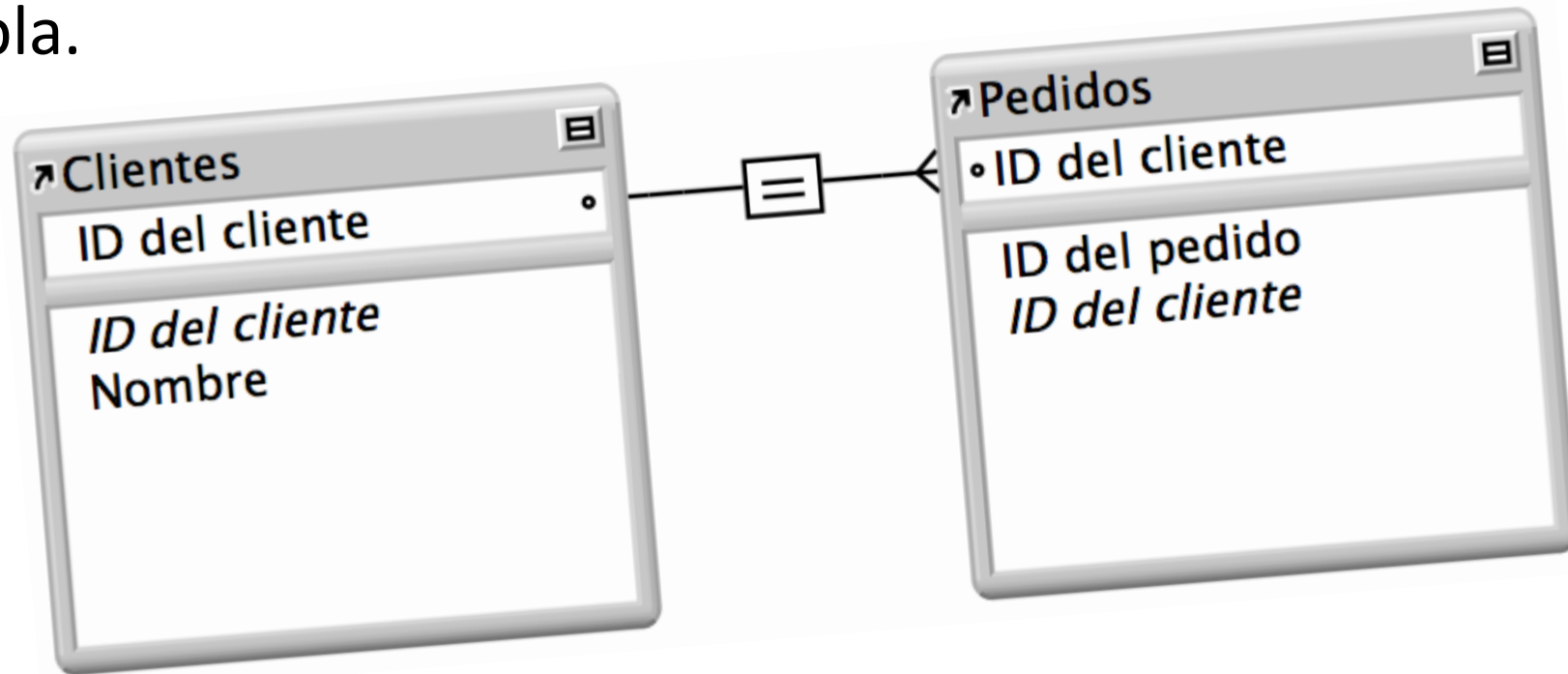
Tipos de Relaciones

De uno a uno: Un registro de una tabla está asociado a uno y solo un registro de otra tabla.



Tipos de Relaciones

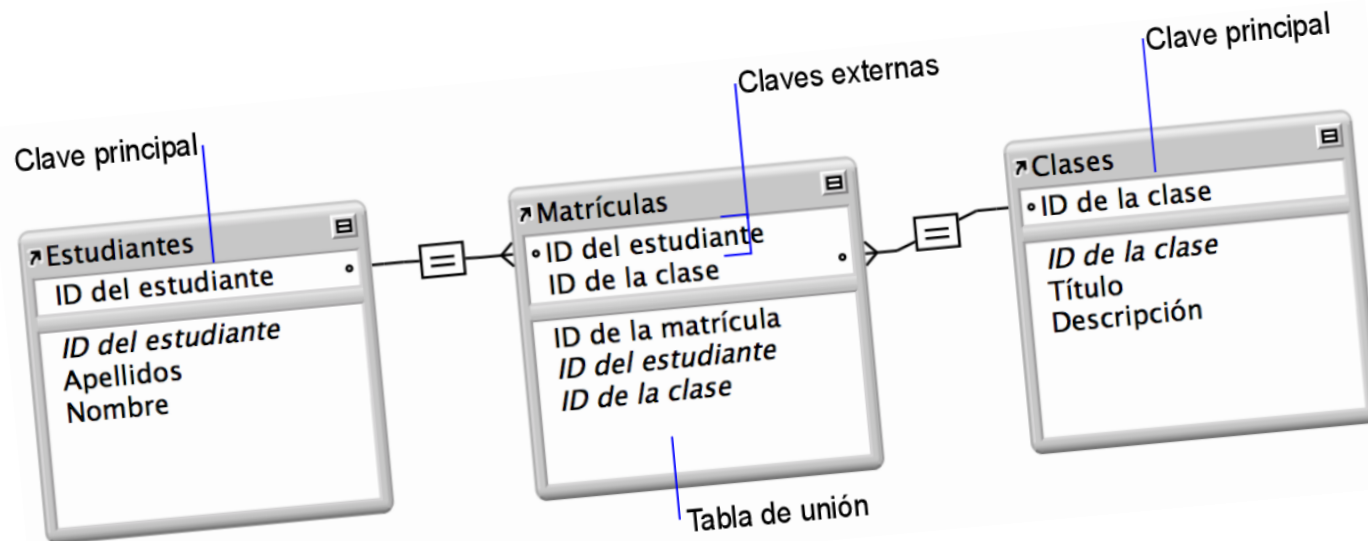
De uno a muchos, o de muchos a uno: Un registro de una tabla puede estar asociado a uno o varios registros de otra tabla. O bien, uno o varios registros de una tabla pueden estar asociados a un registro de otra tabla.



Requisitos para relacionar tablas

- 📶 Indispensable que ambas tengan un campo en común.
- 📶 El campo en común debe ser del mismo tipo de dato. (INTEGER, TEXTO, SMALL INTEGER, etc.)
- 📶 La propiedad Tamaño del campo debe ser igual en ambas tablas.
- 📶 Si el campo en la tabla primaria está definido como de Valor automático en la tabla secundaria debe estar definido como INTEGER.
- 📶 El campo común debe ser Clave principal en la tabla primaria.

Clave primaria (Primary Key):

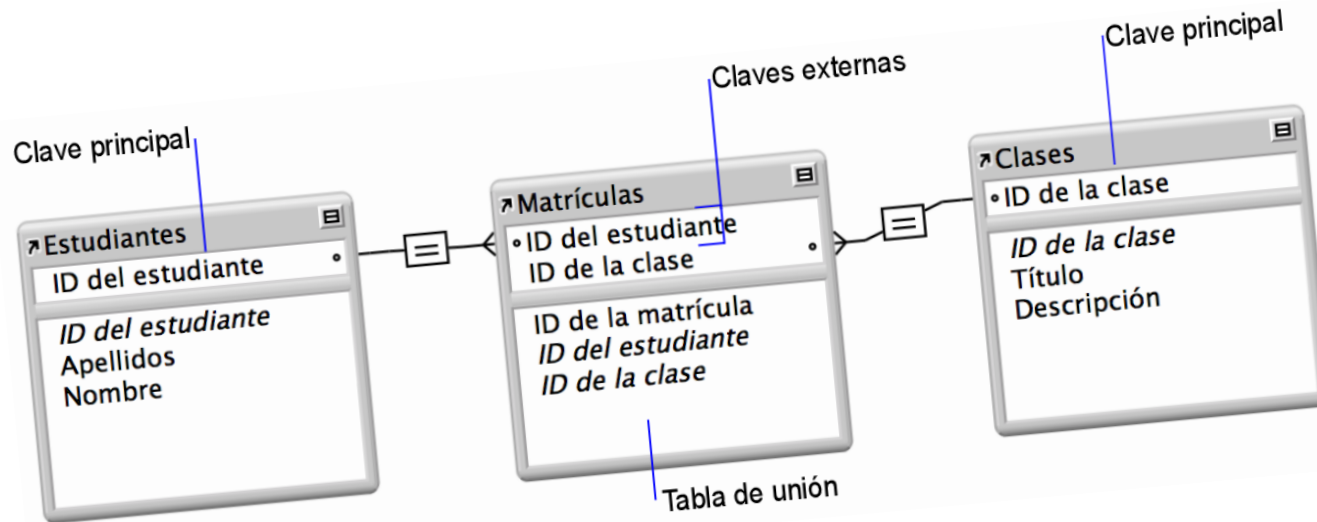


Un campo que se encuentra en la misma tabla que el registro que identifica.

Características

- 📍 Una por tabla.
- 📍 Valor exclusivo
- 📍 Diferente de nulo o vacío

Clave foránea (Foreign key):

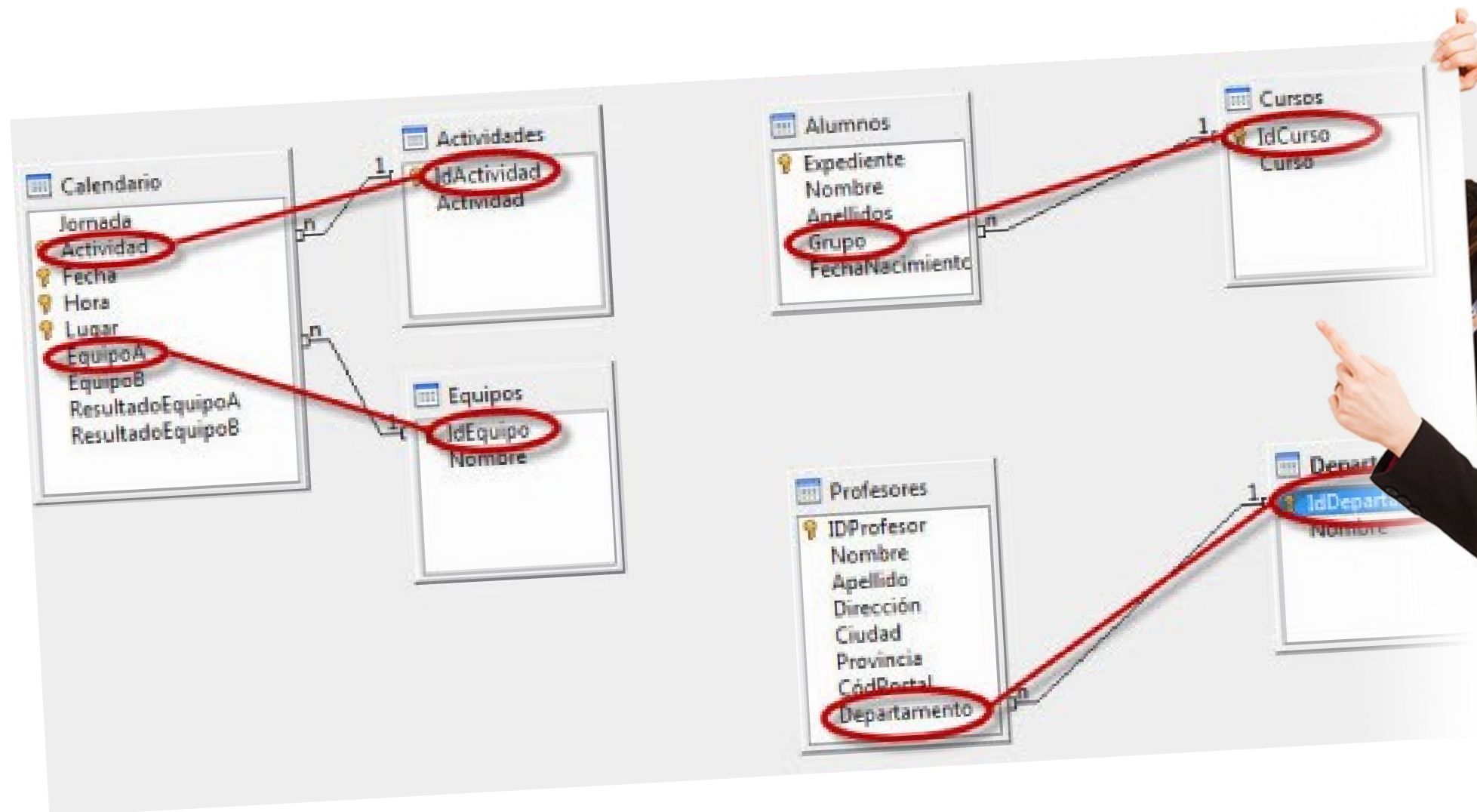


Un campo de una tabla que identifica un registro de otra tabla.

Características:

- 📶 No tienen que ser exclusivas de la tabla.
- 📶 Pueden ser nulo o vacío.
- 📶 Puede existir múltiples claves foráneas en una tabla .

Establecer relaciones entre tablas



Problemas y solución para las relaciones Muchos a muchos

MUCHOS A MUCHOS

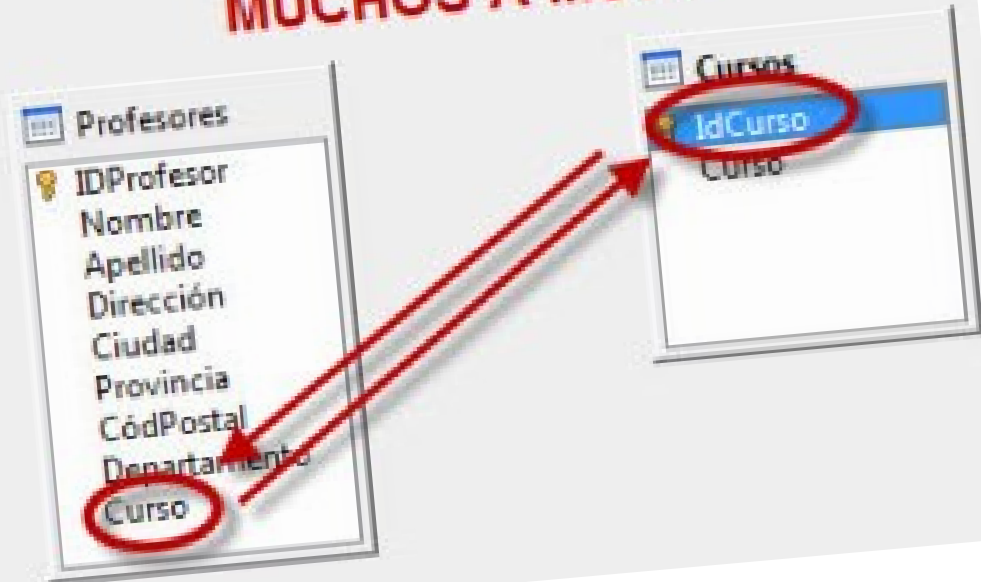


TABLA INTERMEDIA





¿Que es SQL?

SQL

(Structured Query Language)

Lenguaje Estructurado de Consultas



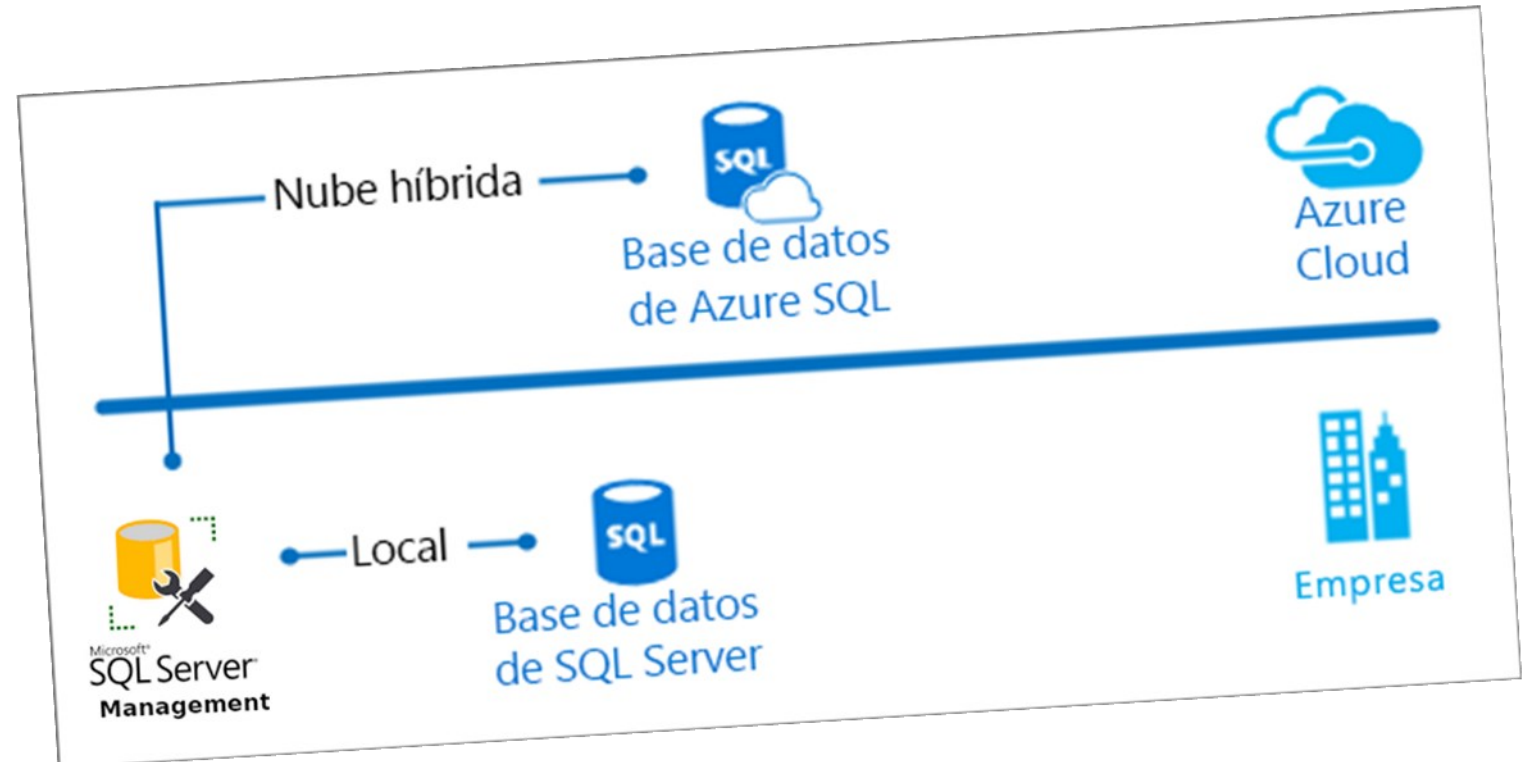
SQL

Es un lenguaje de consulta y programación de bases de datos utilizado para la organización, acceso, consulta y gestión de bases de datos relacionales.



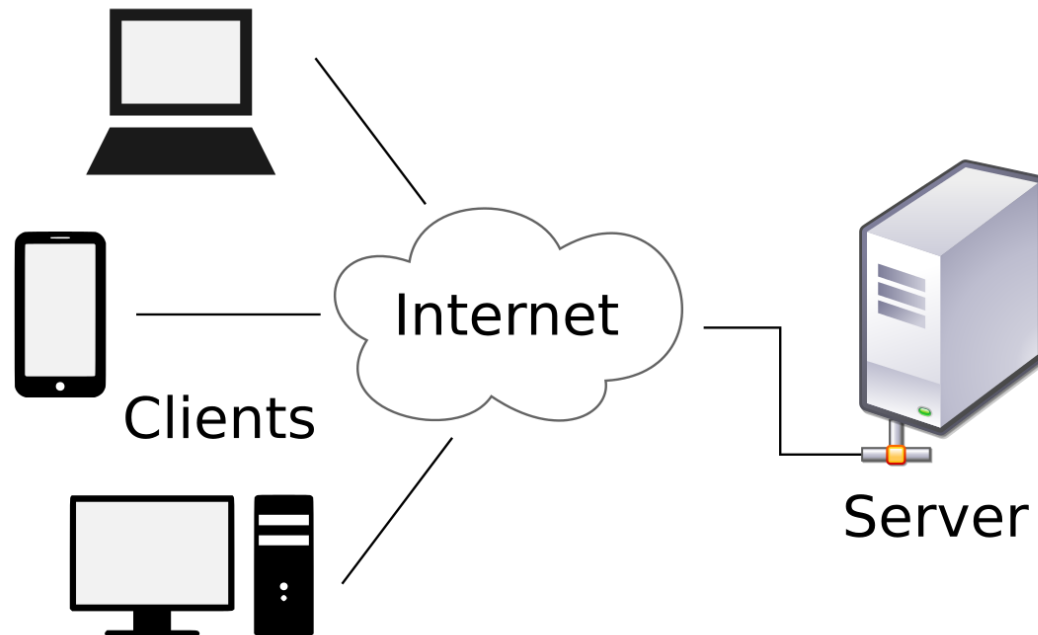
Sistemas de SQL Server

Un Sistema SQL puede implementarse como sistema cliente/servidor o como sistema independiente.



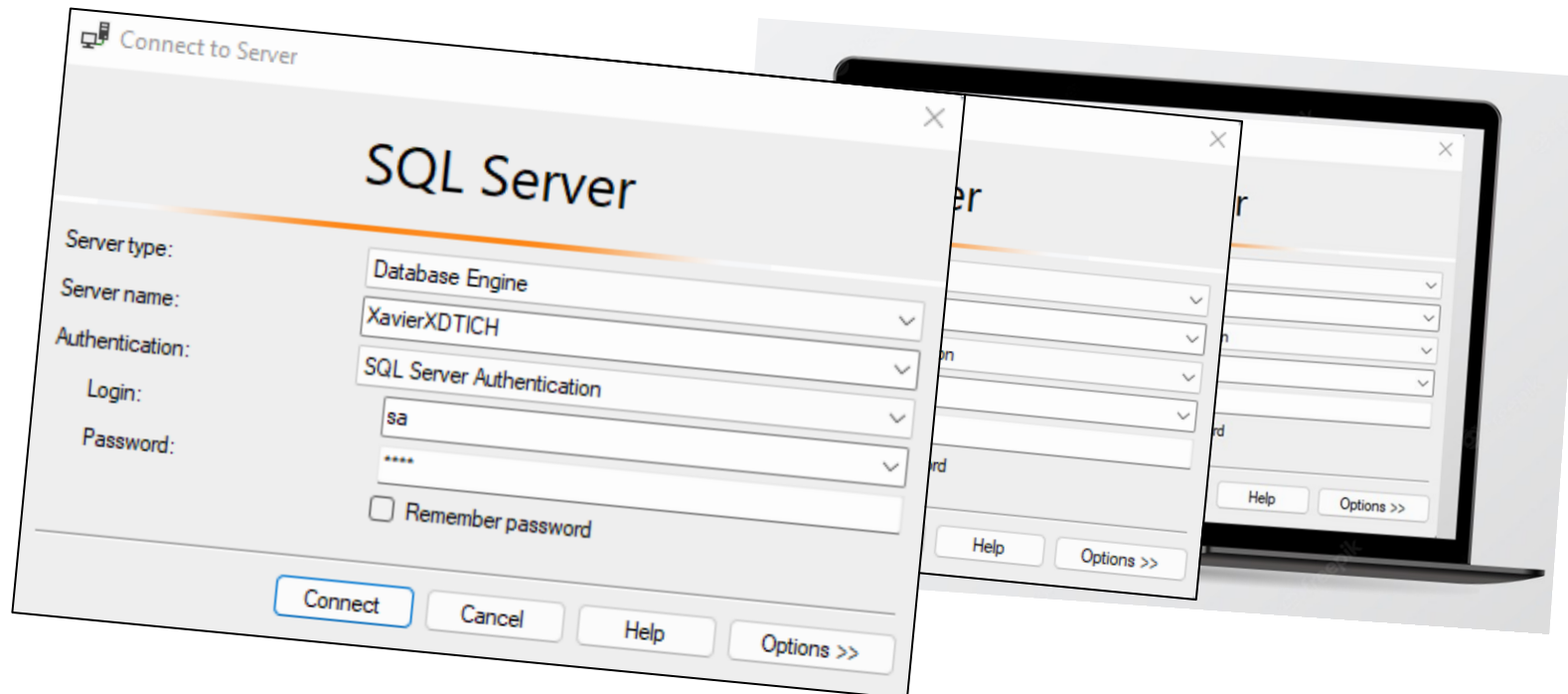
Sistema cliente/servidor

Los usuarios tienen equipos independientes llamados clientes. Los usuarios acceden a la base de datos por medio de aplicaciones en sus equipos cliente.



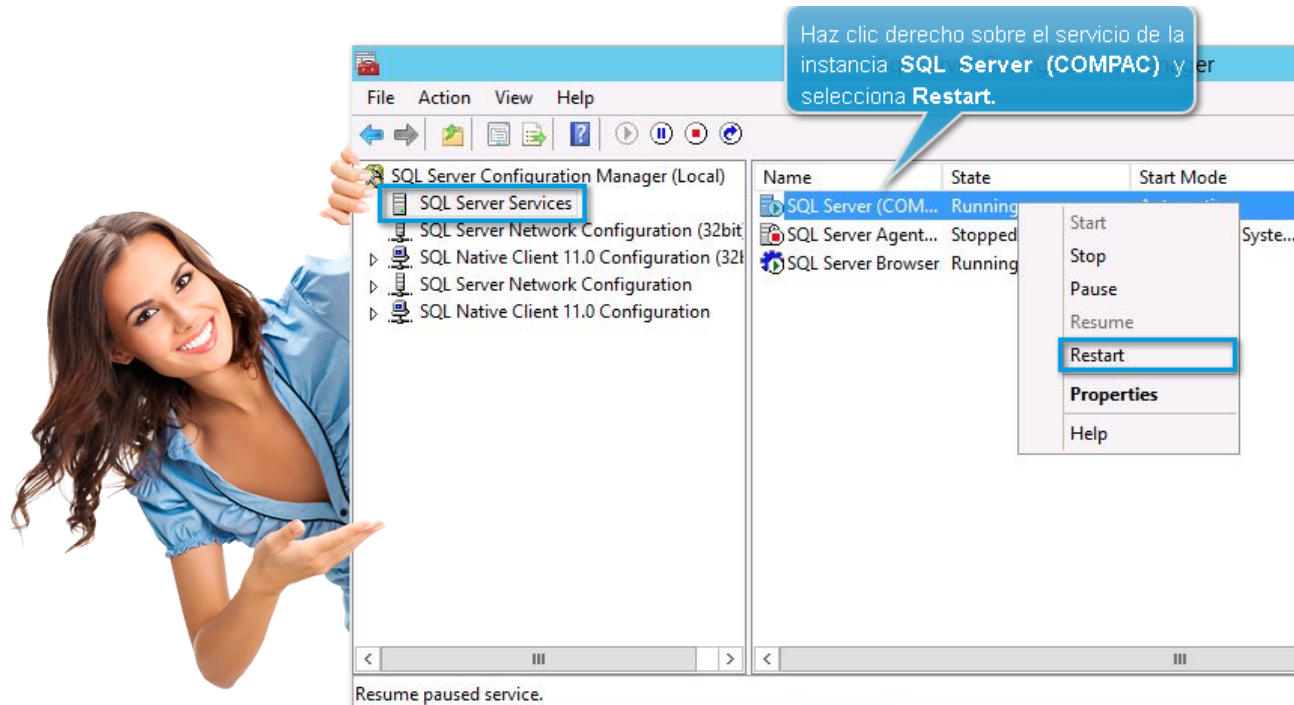
Sistema independiente (Local)

Las aplicaciones cliente se ejecutan en la misma máquina que almacena el motor de SQL Server y las bases de datos, por lo tanto no se realiza ninguna conexión de red del cliente al servidor.



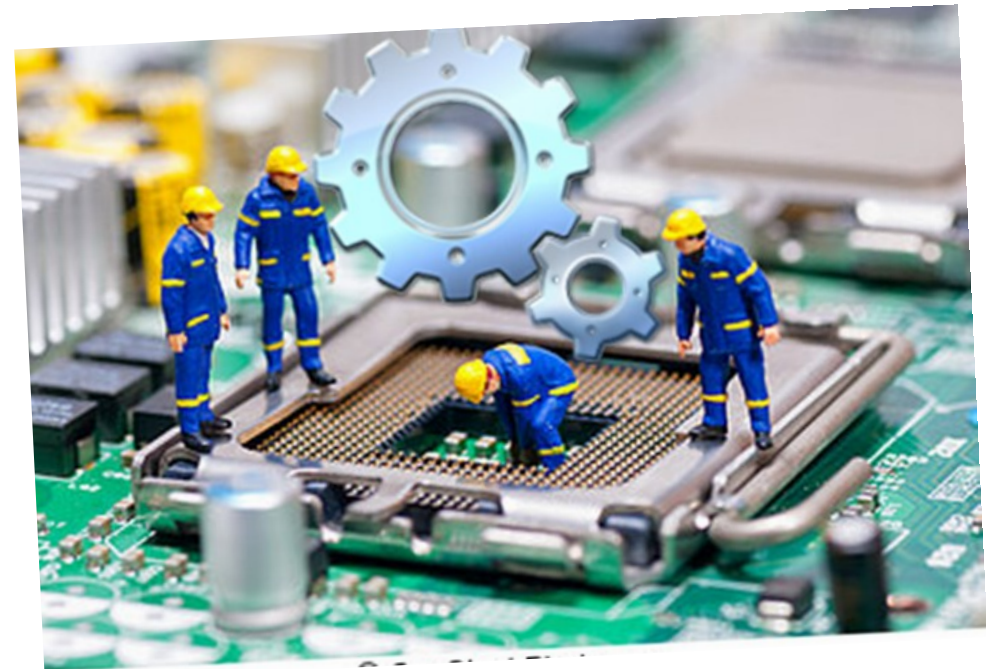
Administración de Servicios SQL Server

- Un servicio es un programa o proceso que lleva a cabo una función específica para dar soporte a otros programas o aplicaciones.
- Al iniciar el equipo, el servicio se inicia de manera automática.




Administración de Servicios SQL Server

📶 El servicio gestiona los archivos de bases de datos, procesa sentencias Transact-SQL, ubica recursos entre conexiones de usuarios y asegura la consistencia de los datos.




Bases de Datos del Sistema

Al instalar SQL Server se crean cuatro bases de datos del sistema: las bases de datos master, tempdb, model y msdb.

 Master.- Registra la información de inicialización de SQL Server y los parámetros de configuración de SQL Server. También registra todas las cuentas de inicio de sesión, la existencia de todas las demás bases de datos y la posición del archivo principal de todas las bases de datos de usuario.


Bases de Datos del Sistema

 Tempdb.- Alberga tablas y procedimientos almacenados temporales. Esta base de datos también se utiliza para otras necesidades de almacenamiento como el ordenamiento de datos.

Bases de Datos del Sistema

📡 Model.- Sirve como plantilla para todas las demás bases de datos que se creen en el sistema, entre ellas tempdb. Esta base de datos debe existir en el sistema por que se utiliza para volver a crear tempdb cada vez que SQL Server se inicia.

Bases de Datos del Sistema

 Msdb.- Contiene tablas que el agente de SQL Server utiliza para la planificación de trabajos y alertas. Esta base de datos también alberga las tablas que se usan para la duplicación.

Transacciones-SQL

Son comandos universales SQL que permiten realizar petición a un servidor de BD por medio de otras aplicaciones.

DDL(Data Definition Language)
Lenguaje de Definición de Datos

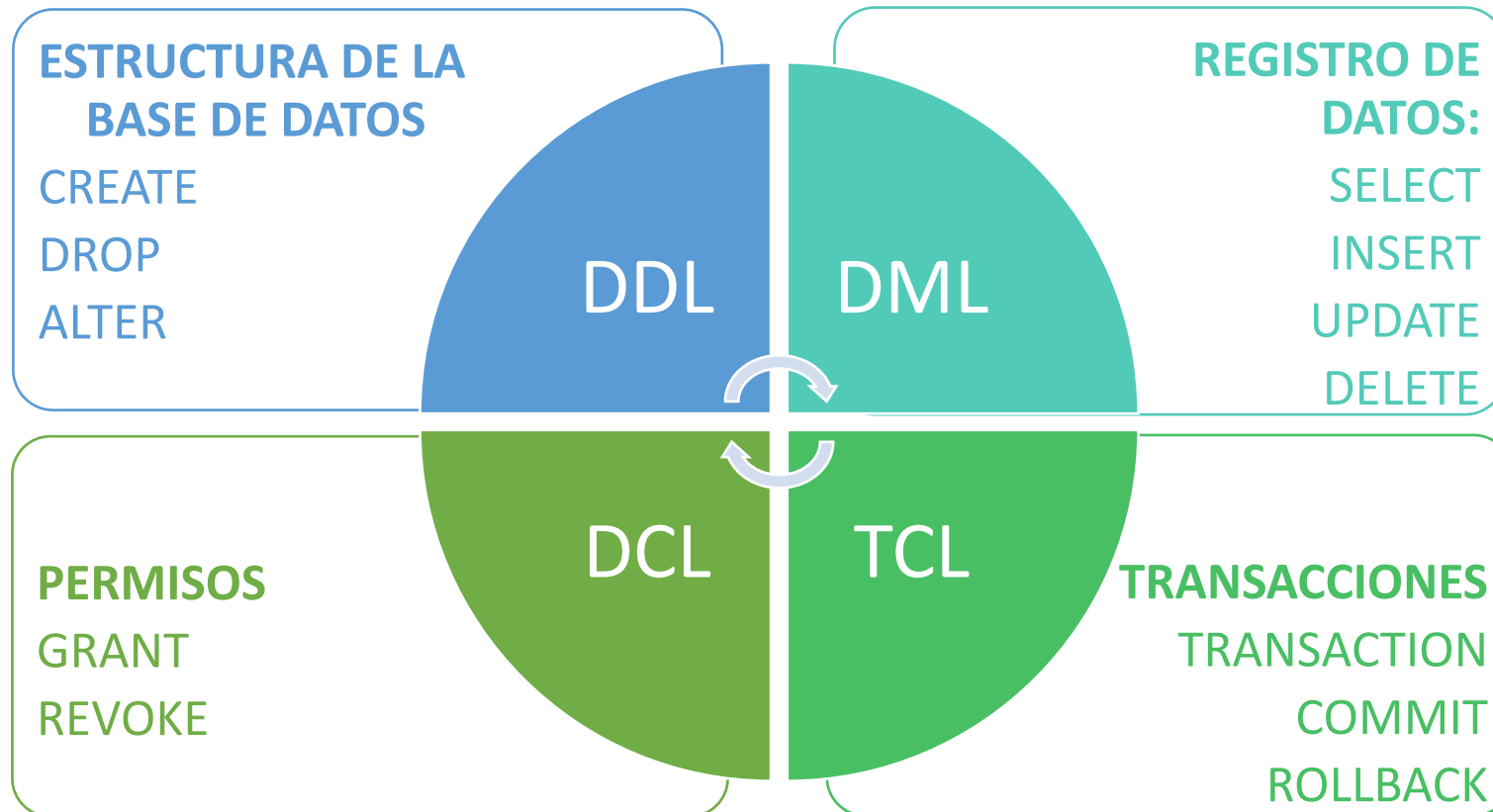
DML(Data Manipulation Language)
Lenguaje de Manipulación de Datos

DCL(Data Control Language)
Lenguaje de Control de Datos

TCL(Transactional Control Language)
Lenguaje de Control Transaccional

Transacciones-SQL

Se agrupan en dos categorías principales o sub-lenguajes DDL Y DML:



Diferencias

DDL (Lenguajes de Definición de Datos)	DML (Lenguaje de Manipulación de Datos)
Trabaja sobre su estructura	Trabaja sobre los registros
Trabaja a nivel de tablas	Trabaja a nivel de registros
Trabaja sobre la metadata	Trabaja sobre la data
Manipula la descripción de los objetos de la Base de Datos	Manipula la información que se almacena dentro de una o más tablas
Trabaja sobre la estructura de esa información	Trabaja sobre la información

Crear una Base de Datos (CREATE)

 Comando T-SQL:

- CREATE DATABASE

 Sintaxis:

- CREATE DATABASE [Nombre de BD]

Utilizar una BD

📡 Para realizar una acción dentro de una Base de Datos, primero hay que decirle a SQL Server que queremos trabajar con esa Base de Datos, para esto se utiliza el siguiente comando

📡 Sentencia SQL:

- USE

📡 Sintaxis:

- USE Nombre de la BD

Ejemplo:
USE Animales

SQLQuery1.sql - TICHINSTRUCTOR.InstitutoTich (sa (64)) - Microsoft SQL Server Man...

File Edit View Query Project Tools Window Help

InstitutoTich Execute

Object Explorer

Connect

TICHINSTRUCTOR (SQL Server 15.0.2000.5)

Databases

System Databases

Database Snapshots

EjerciciosTich

InstitutoTich

Database Diagrams

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.Alumnos

dbo.AlumnosBaja

dbo.CatCursos

dbo.Cursos

dbo.CursosAlumnos

dbo.CursosInstructores

dbo.Estados

dbo.EstadosAlumnos

dbo.Instructores

dbo.TablaISR

Views

External Resources

Synonyms

Programmability

Service Broker

Storage

Security

SQLQuery1.sql - TICHINSTRUCTOR.InstitutoTich (sa (64))

```
USE [InstitutoTich]
GO

SELECT [id]
,[nombre]
,[primerApellido]
,[segundoApellido]
,[telefono]
```

176 %

Results Messages

	id	nombre	primerApellido	segundoApellido	correo	telefono	fechaNacimiento	curp
1	1	Marcelo Isai a	García	mirón	marcelo@outlook.com	9911362600	1988-04-18	DIAE
2	2	Oliver Alexis	Martínez	Estudillo	alexis@gmail.com	8897877417	1996-04-18	DIAE
3	3	Oscar	Mendoza	garcía	omscar@outlook.es	7711589568	1994-04-07	RUVJ940
4	4	Edgar	Martínez	Espinoza	edgargmail.com	5528356144	1996-05-23	DOML960323HM
5	5	Rodrigo	Tolentino	Martínez	rodrigo@gmail.com	4421436224	1998-03-13	TOMR980313HI
6	6	Jesiel	García	Pérez	jesiel@gmail.com	3317901341	1990-11-08	GAPJ901108HH
7	7	Christian Josue	Gonzalez	Lozano	christian@gmail.com	4922153353	1996-06-19	GOLC960619HZ
8	8	Luis Enrique	Lopez	Cruz	luis@gmail.com	2235700644	1997-07-15	LOCL970715HG
9	9	Rolando	Marquez	Hernandez	rolando@gmail.com	1168329969	1997-03-08	MAHR97030815
10	10	Jesús Yotecatl	Miranda	Espinosa	jesus@gmail.com	2213335247	1997-06-14	MIEJ970614HMC
11	11	Cecilia	Cruz	Luna	cecilia@outlook.com	3317052376	1997-08-08	CULC970808MP

Query executed successfully.

TICHINST

Ready

Ln 20 Col 1 Ch 1 INS

2
Crear Consulta

1
Selección de BD

4
Ejecutar consulta

3
Escribir consulta

6
Resultado de Consulta

5
Verificar si se ejecuto bien



1. Creación de Tablas

Creación de una Tabla

Comando T-SQL: DDL

- CREATE TABLE

Sintaxis:

- CREATE TABLE *Nombre_Tabla* (
Nombre_Campo1 Tipo_Dato[(Tamaño)] [Restricciones –NULL | NOT NULL]
 [DEFAULT Valor],
Nombre_Campo2 Tipo_Dato[(Tamaño)] [Restricciones – NULL | NOT NULL]
 [DEFAULT Valor],
 [restricciones tabla – PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK....]
 etc....)
- * Solo es necesario definir el tamaño cuando se utilizan los Tipos de Datos Char y Varchar.

Ejemplo:

```
CREATE TABLE PRODUCTOS (
codigo_producto INTEGER IDENTITY(1,1),
nombre_producto VARCHAR(20) UNIQUE NOT NULL,
tipo VARCHAR(20),
descripcion VARCHAR(50),
precio DECIMAL DEFAULT 1.0,
id_fabrica INTEGER DEFAULT NULL,
PRIMARY KEY (codigo_producto),
FOREIGN KEY (id_fabrica) REFERENCES FABRICAS(id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT precio CHECK (precio>5)
);
```

Marcadores y restricciones

- 📡 **IDENTITY**(valor inicial, incremento) - Autonumérico, secuencial que va asignando el entero siguiente al máximo valor almacenado para el campo.
- 📡 **DEFAULT** val- Establece un valor por defecto al campo.
- 📡 **NOT NULL** - No puede contener valores nulos.
- 📡 **PRIMARY KEY**- El campo es la clave primaria no compuesta.
- 📡 **REFERENCES**- tabla [(campo)] - Clave ajena no compuesta.
- 📡 **CHECK** (condicional) - El campo debe cumplir una condición. Tanto a las claves primarias y ajenas como a los chequeos de condición se les puede anteponer la partícula.

Marcadores y restricciones

 **CONSTRAINT** nombre, para nombrar la restricción.

 Integridad referencial en cascada

Opción de cascada	Comportamiento de UPDATE	Comportamiento de DELETE
NO ACTION (valor predeterminado)	Produce error, deshace operación	
CASCADE	Actualizar claves externas en las tablas referenciadas.	Eliminar filas en las tablas Referenciadas.
SET NULL	Establecer en NULL las claves externas en las tablas de referencia	
SET DEFAULT	Establecer valores predeterminados las claves externas en las tablas de referencia.	

Comandos ALTER

Comando T-SQL: DDL

- ALTER TABLE

Sintaxis:

- Para agregar un nuevo campo a la tabla
 - `ALTER TABLE FABRICAS ADD proveedor VARCHAR(50);`
- Para eliminar un campo de la tabla
 - `ALTER TABLE FABRICAS DROP descripcion;`
- Adición de restricciones, claves
 - `ALTER TABLE PRODUCTOS ADD CONSTRAINT FK_idfabrica FOREIGN KEY (Id_Fabrica) REFERENCES FABRICAS(Id);`

Ejemplos:

📡 Para agregar un nuevo campo a la tabla

- `ALTER TABLE FABRICAS ADD proveedor VARCHAR(50);`

📡 Para eliminar un campo de la tabla

- `ALTER TABLE FABRICAS DROP descripcion;`

📡 Adición de restricciones, claves

- `ALTER TABLE PRODUCTOS ADD CONSTRAINT id_fabrica`
- `FOREIGN KEY (fabrica) REFERENCES FABRICAS;`