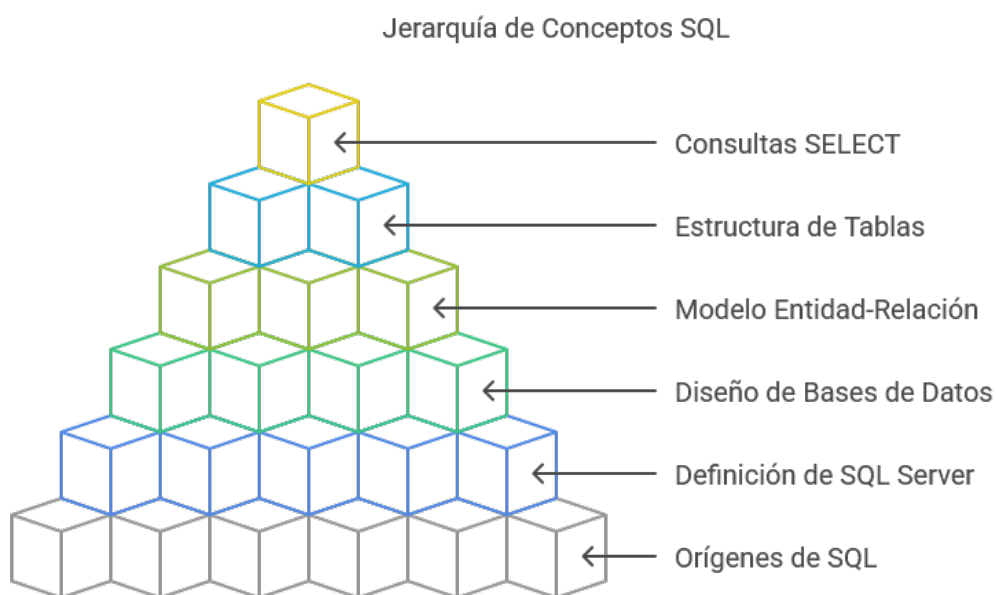


Diseño de Bases de Datos Relacionales

Introducción:

Este documento presenta un resumen de los conceptos clave SQL.

Se abarcan desde los orígenes del lenguaje SQL y la definición de SQL Server, hasta los fundamentos del diseño de bases de datos relacionales, incluyendo el modelo Entidad-Relación, la estructura de tablas, las restricciones de datos y las consultas básicas utilizando la sentencia SELECT.



Temas Principales y Ideas Clave:

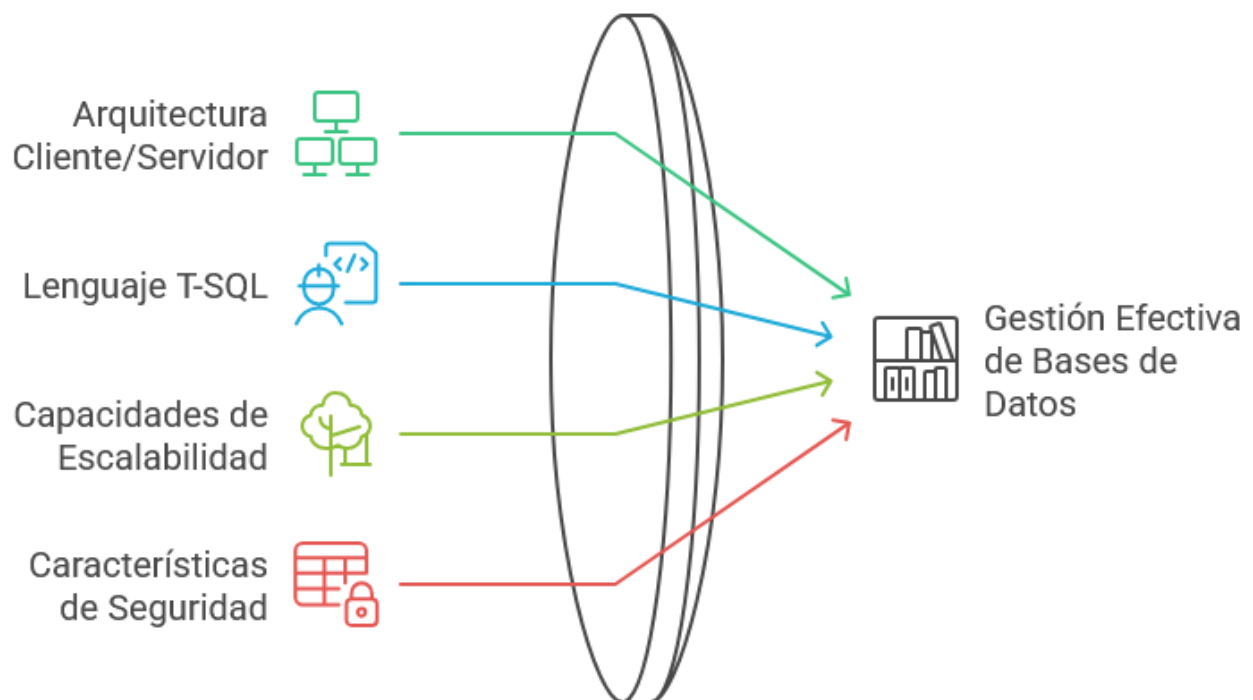
1. Orígenes e Historia de SQL:

- SQL se consolidó como el lenguaje estándar para bases de datos relacionales.
- El lenguaje original, SEQUEL, fue desarrollado en IBM a principios de la década de 1970 como parte del proyecto System R.
- **Cita:** "La versión original se desarrolló en el laboratorio de investigación de San Jose, California (San Jose Research Center) de IBM, este lenguaje originalmente denominado Sequel, se implementó como parte del proyecto System R, a principios de 1970 [McJones97]."
- A lo largo del tiempo, SQL evolucionó y fue estandarizado por ANSI e ISO, dando lugar a normas como SQL-86, SQL-89 y la actual SQL-92.
- **Cita:** "En 1986, ANSI (American National Standards Institute, Instituto Nacional Americano de Normalización) e ISO (International Standards Organization, Organización Internacional de Normalización) Publicaron una norma de SQL denominada SQL-86."
- Es importante tener en cuenta que no todas las implementaciones de SQL son compatibles con la norma más reciente (SQL-92), siendo algunas compatibles solo con SQL-89.

2. ¿Qué es SQL Server?

- SQL Server se define como un sistema administrador para bases de datos relacionales basado en la arquitectura cliente/servidor.
- **Cita:** "SQL Server es un sistema administrador para bases de datos relacionales basadas en la arquitectura cliente/servidor."
- Utiliza Transact-SQL (T-SQL) como su lenguaje propietario para la comunicación entre cliente y servidor. T-SQL se basa en el estándar SQL utilizado por la mayoría de las bases de datos relacionales.
- **Cita:** "Transact-SQL es el lenguaje que emplea para mandar peticiones entre el cliente y el servidor. Es un lenguaje exclusivo de SQL Server, pero basado en el lenguaje SQL estándar..."
- El propósito principal de SQL Server es analizar y administrar datos, proporcionando escalabilidad, disponibilidad y seguridad para aplicaciones empresariales.

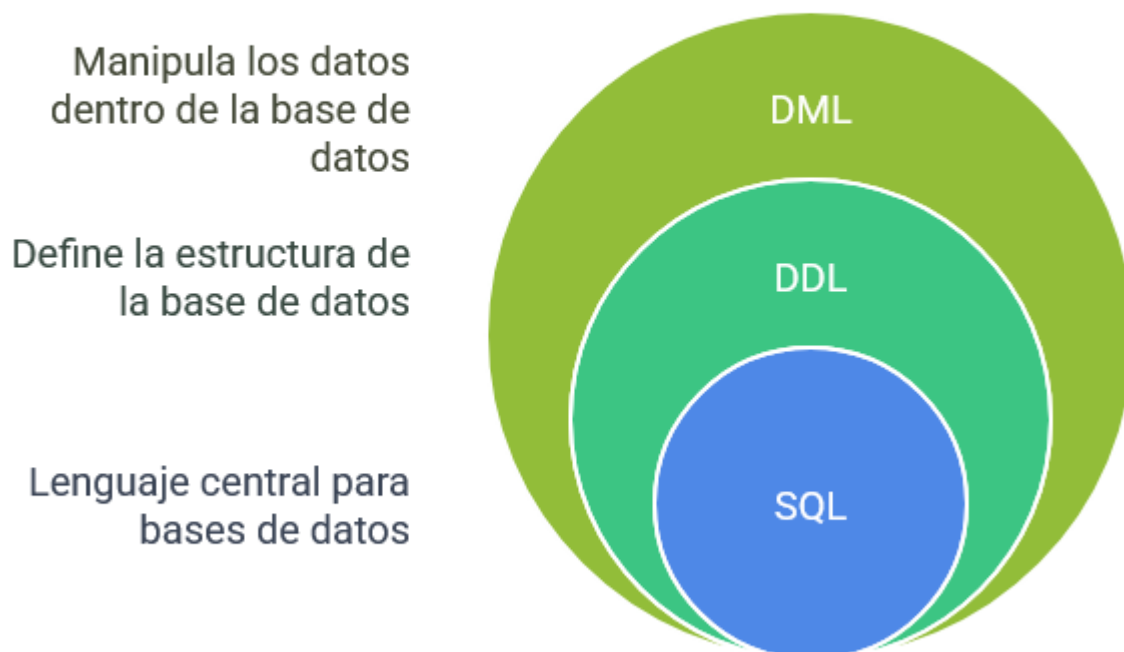
Logros de SQL Server



3. Características de SQL:

- SQL proporciona dos sublenguajes distintos:
- **Lenguaje de Definición de Datos (DDL):** Se utiliza para especificar el esquema relacional, es decir, la estructura de la base de datos (tablas, relaciones, reglas). La compilación de estas instrucciones genera metadatos almacenados en el diccionario de datos o catálogo del sistema.
- **Cita:** "Un esquema de bases de datos se representa mediante un sublenguaje especial llamado lenguaje de definición de datos."
- **Lenguaje de Manipulación de Datos (DML):** Se emplea para realizar consultas y actualizaciones de los datos almacenados en la base de datos. Esto incluye la recuperación, inserción, eliminación y modificación de información.
- **Cita:** "Un D.M.L. es un sublenguaje de consulta y manipulación de datos. Se entenderá por manipulación de datos a: Recuperación de Información. Inserción de nueva Información. Eliminación (Borrado) de información existente. Modificación de Información Almacenada."

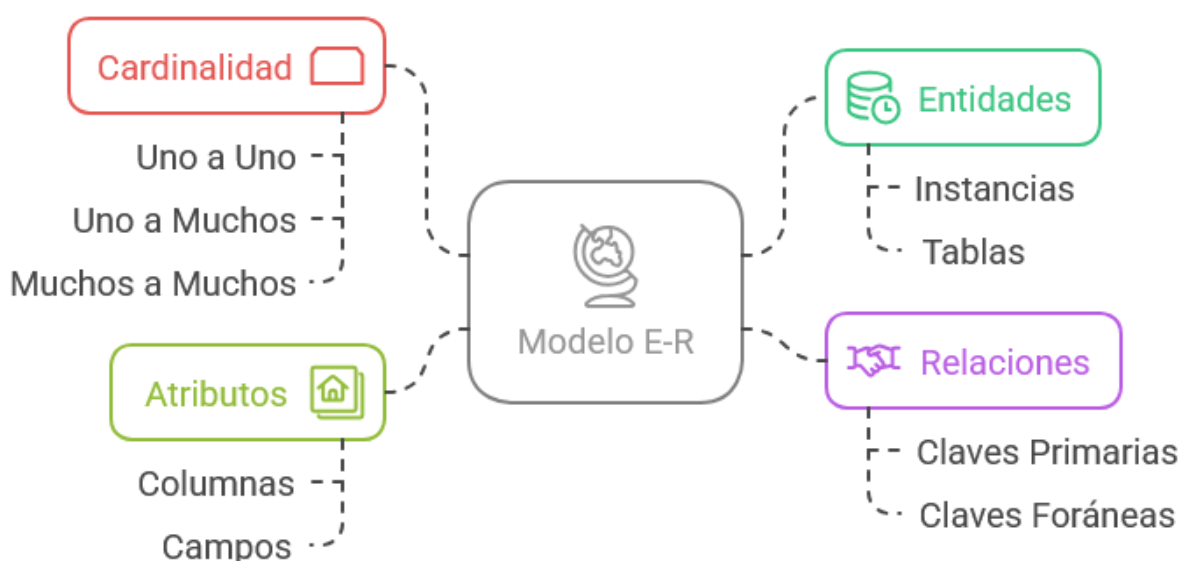
Estructura de SQL



4. Modelo Entidad-Relación (E-R):

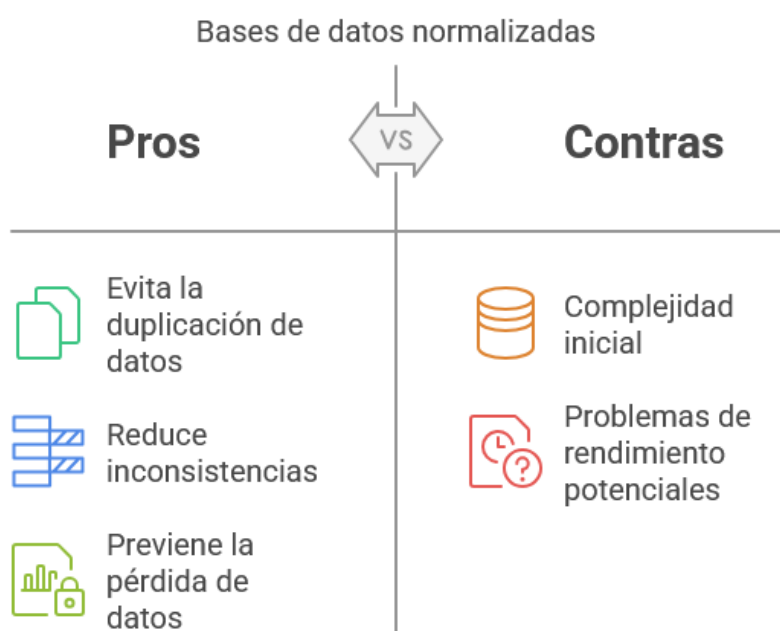
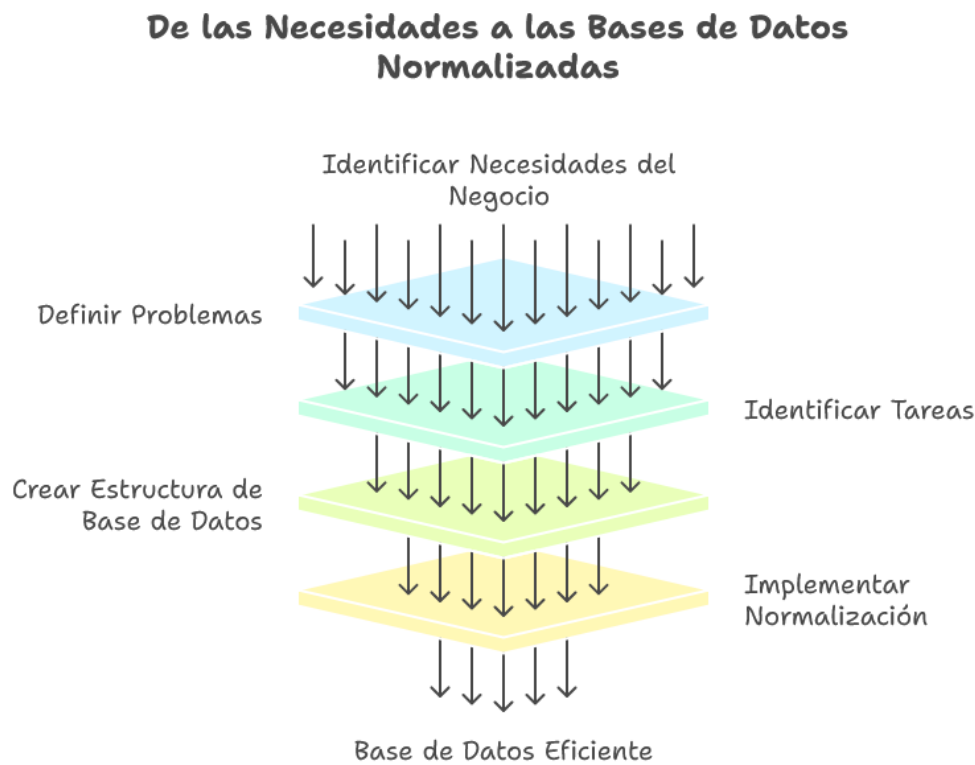
- Antes de la creación física de las tablas, se debe realizar un modelo de datos. El modelo E-R es un modelo conceptual para el diseño de bases de datos que representa una parte del mundo real en términos de entidades y las relaciones entre ellas.
- **Cita:** "Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas se debe realizar un modelo de datos."
- **Entidad:** Un "objeto" discreto sobre el que se almacena información. Cada instancia de una entidad se llama instancia. En la base de datos, las entidades se modelan como tablas.
- **Cita:** "Una entidad es cualquier 'objeto' discreto sobre el que se tiene información."
- **Relación:** Describe la interdependencia entre una o más entidades. Las relaciones se implementan utilizando claves primarias y foráneas para mantener la integridad referencial.
- **Cita:** "Una relación describe cierta interdependencia (de cualquier tipo) entre una o más entidades."
- **Cardinalidad de las Relaciones:** Define el número de instancias de una entidad que pueden estar relacionadas con otra entidad:
 - Uno a uno.
 - Uno a muchos.
 - Muchos a muchos.
- **Atributos:** Propiedades de las entidades que se desea almacenar. En el modelo de base de datos, los atributos se almacenan como columnas o campos de una tabla.
- **Cita:** "Las entidades tienen atributos. Un atributo de una entidad representa alguna propiedad que nos interesa almacenar."

Modelo Entidad-Relación en Diseño de Bases de Datos



5. Planeamiento del Diseño Lógico de la Base de Datos:

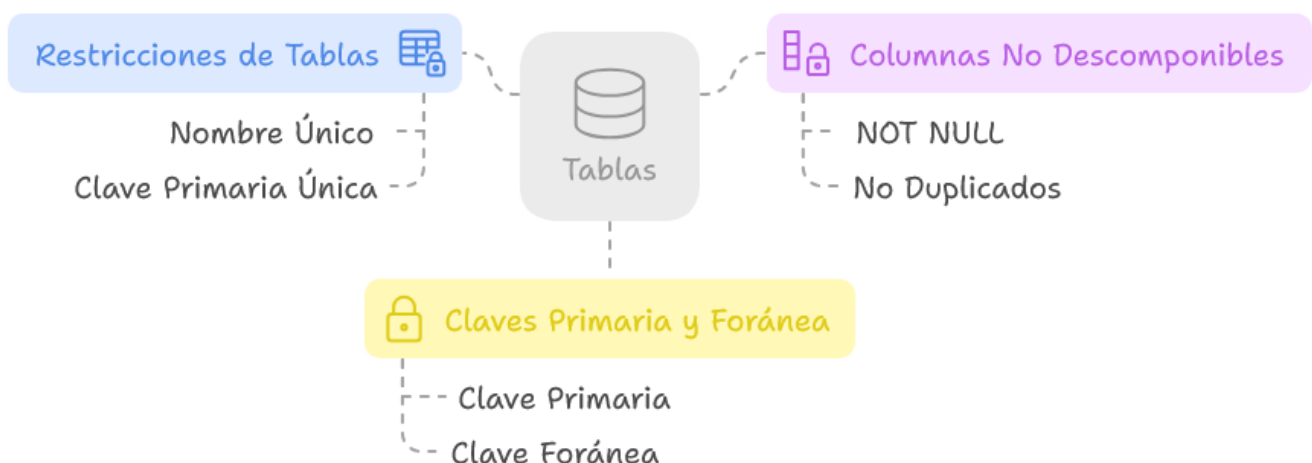
- Es crucial determinar las necesidades del negocio y los usuarios, identificando los problemas a resolver y las tareas a completar.
- Se recomienda crear bases de datos normalizadas para evitar la duplicación de información, inconsistencias y problemas de pérdida de datos.



6. Modelado de Elementos de Datos:

- **Tablas:** Son los objetos principales de almacenamiento de datos en SQL. Consisten en una estructura de columnas que almacenan información relacionada en filas o registros. Cada tabla tiene un nombre único dentro de la base de datos.
- **Cita:** "SQL emplea tablas como objetos de almacenamiento de datos, que los usuarios manipulan a través de sus aplicaciones."
- **Restricciones de Tablas:** Los nombres de las tablas deben ser únicos en la base de datos.
- Los nombres de las columnas deben ser únicos dentro de la tabla.
- No puede haber dos registros con el mismo valor de la clave primaria.
- **Columnas No Descomponibles:** Contienen información indivisible, lo que facilita la actualización, consulta y mantenimiento de la integridad de los datos.
- **Cita:** "Son aquellas columnas que contienen cierta información que no puede ser en dos o más columnas."
- **Restricciones en Columnas: NOT NULL:** Asegura que una columna no pueda contener valores nulos.
- **No Duplicates:** (Implícito en claves primarias o índices únicos) Evita valores duplicados en una columna.
- **Clave Primaria (PRIMARY KEY):** Una columna o conjunto de columnas que garantiza la unicidad de cada fila en una tabla y no permite valores nulos. Solo puede haber una clave primaria por tabla.
- **Cita:** "La Clave Primaria es una columna o un grupo de columnas que fuerzan la integridad de los datos en la tabla, asegurándose que cada fila en la tabla es la única."
- **Clave Foránea (FOREIGN KEY):** Una columna o conjunto de columnas que referencia la clave primaria de otra tabla (o de la misma tabla). Se utiliza para establecer y mantener la integridad referencial entre tablas. SQL Server valida los datos almacenados en una clave foránea.
- **Cita:** "La Clave Foránea referencia a la clave primaria de una tabla."

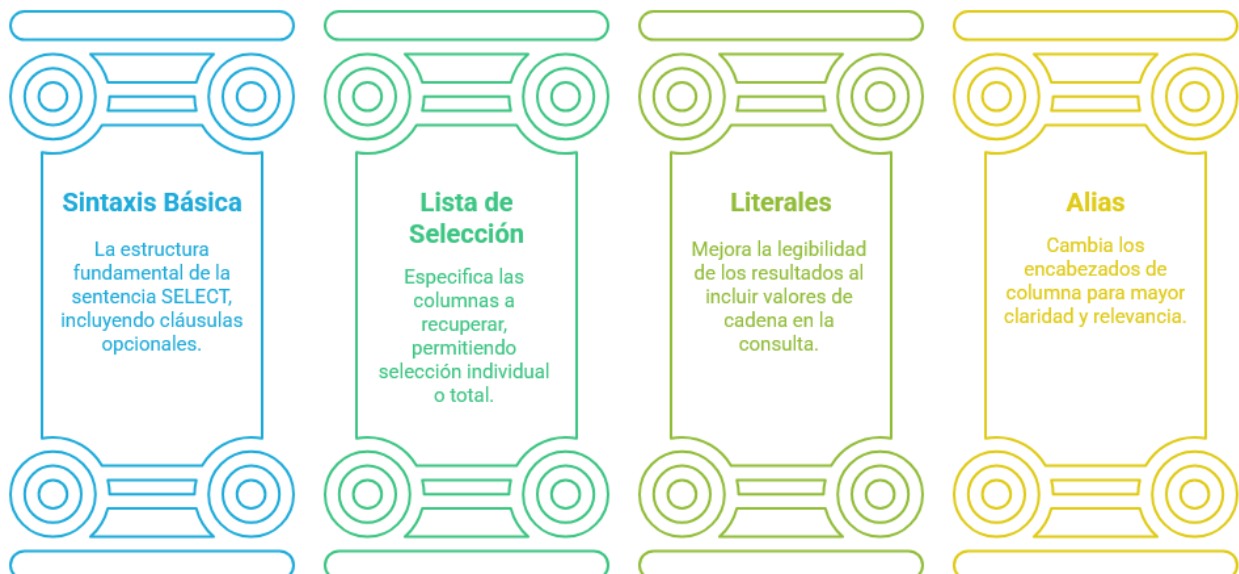
Estructura y Restricciones de Tablas en SQL



7. Consultas Básicas - Sentencia SELECT:

- La sentencia SELECT se utiliza para recuperar información de la base de datos, permitiendo la selección de filas y columnas de una o varias tablas.
- **Cita:** "Recupera la información de la base de datos y permite la selección de una o más filas o columnas de una o muchas tablas"
- **Sintaxis Básica:** SELECT select_list
- FROM table_source
- [WHERE search_condition]
- [GROUP BY group_by_expression]
- [HAVING search_condition]
- [ORDER BY order_expression [ASC | DESC]];
- La select_list especifica las columnas a recuperar. Se pueden usar comas para separar los nombres de las columnas o un asterisco (*) para seleccionar todas las columnas.
- El orden en que se listan las columnas en la sentencia SELECT determina el orden en que aparecerán en el resultado.
- Se pueden incluir **literales** (cadenas de caracteres) en la consulta para hacer los resultados más legibles.
- **Ejemplo:** SELECT NOMBRE, 'Telefono = ', TELEFONO FROM CLIENTES
- Los **encabezados de las columnas** resultantes pueden cambiarse utilizando **alias** con las palabras clave AS o simplemente colocando el alias después del nombre de la columna, o utilizando el formato alias = columna.
- **Ejemplo:** SELECT au_lname AS apellido, au_fname AS nombre FROM authors

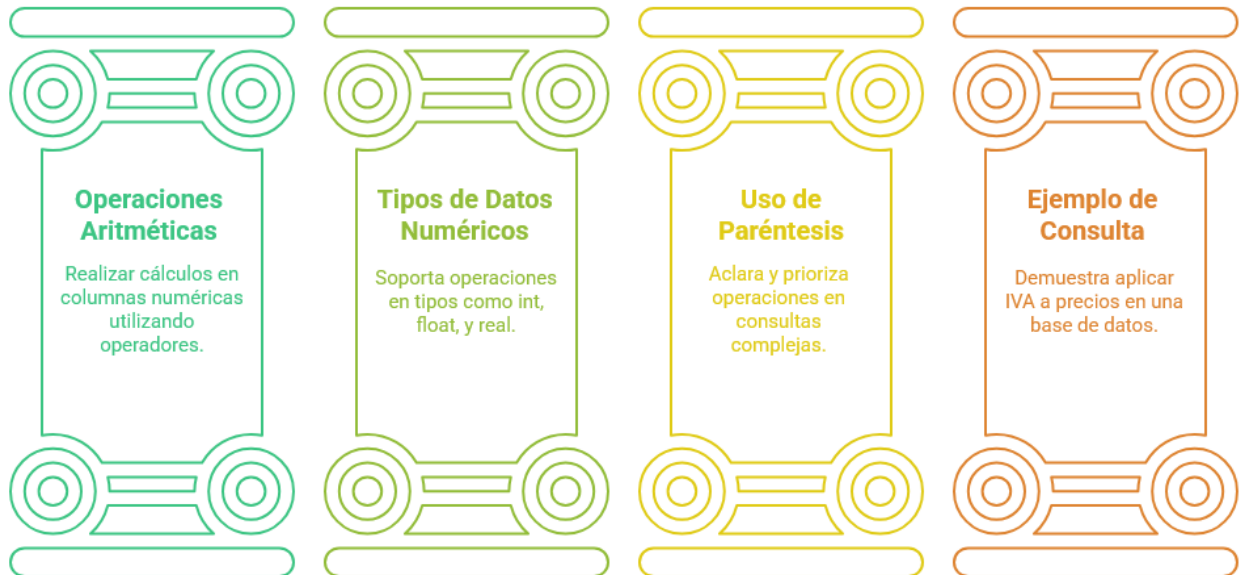
Dominando la Sentencia SELECT



8. Manipulación de Datos (Ejemplos Básicos):

- Se pueden realizar operaciones aritméticas en columnas de tipo numérico (int, float, real, etc.) utilizando operadores como *, +, -, /. Es recomendable usar paréntesis para clarificar operaciones complejas.
- **Ejemplo:** `SELECT price, (price * 1.21) AS price_with_iva, title FROM titles`

Optimización de Cálculos Numéricos en Consultas SQL



Conclusiones:

El material revisado proporciona una introducción sólida a los conceptos fundamentales de SQL Server y el diseño de bases de datos relacionales. Se destaca la importancia de comprender la historia y evolución de SQL, la arquitectura de SQL Server, la distinción entre DDL y DML, y los principios del modelado de datos utilizando el modelo Entidad-Relación. Asimismo, se introducen los elementos básicos de la estructura de tablas, las restricciones de datos para garantizar la integridad y las consultas elementales con la sentencia SELECT. Estos conceptos constituyen la base para el desarrollo y la gestión eficiente de bases de datos relacionales utilizando SQL Server.

Fundamentos de SQL Server y Diseño de Bases de Datos

