

# Modelo Relacional

---

Con un presente dominado por los avances de la IA generativa y el análisis de inmensos volúmenes de datos para crear conocimiento y decisión, en los más variados ámbitos...



*¿Cómo un modelo creado en los años 70 ha sentado las bases para el impacto más significativo en el mundo de la informática de la gestión de datos?*

# Modelo Relacional

---

Con un presente dominado por los avances de la IA generativa y el análisis de inmensos volúmenes de datos para crear conocimiento y decisión, en los más variados ámbitos...



*¿Cómo un modelo creado en los años 70 ha sentado las bases para el impacto más significativo en el mundo de la informática de la gestión de datos?*

Objetivo: Comprender el Modelo Relacional, crear estructuras de datos básicas como tablas con claves primarias y foráneas, y desarrollar habilidades en consultas básicas utilizando sentencias SELECT con las cláusulas FROM y WHERE.

---

*...descubriendo el poder de los datos*



# Modelo Relacional

## Modelo de Datos Jerárquico

Datos se representan en una estructura de árbol con niveles interconectados.



El álgebra relacional se basa en la teoría de conjuntos y proporciona una forma estandarizada y poderosa de realizar consultas y manipulaciones de datos en bases de datos relacionales.

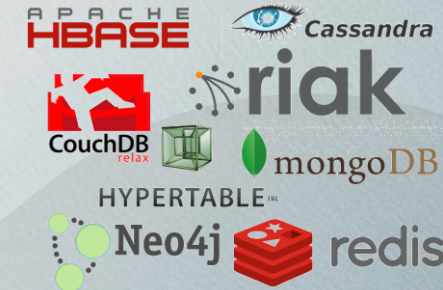


## BIG DATA

Grandes volúmenes de datos, estructurados / no estructurados. Escalabilidad, disponibilidad y procesamiento rápido.



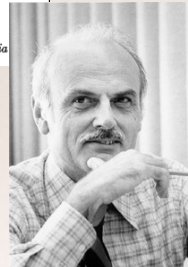
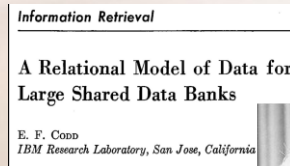
NoSQL



**SGBD Distribuidos – Cliente/Servidor**  
Red de computadoras, procesamiento de datos, acceso remoto.



## SISTEMAS RELACIONALES



Edgar Frank Codd

1980 a 1990

**Expansión de la Funcionalidad**  
Soporte transacciones, integridad de los datos, seguridad, optimización consultas. Adopción en organizaciones



1970 a 1980

**Modelo Relacional**  
Representa los datos en tablas bidimensionales (filas y columnas).

DBTG

**Data Base Task Group**  
Define estándares para la creación de bases de datos y manejo de datos



**Modelo en Red**  
Representaba relaciones de datos más complejas. Primera generación de SGBD. IDS – Integrated Data Store

**SGBD**  
Sistemas de Gestión de Bases de Datos



**Sistema de Ficheros**  
Conjunto de archivos de datos y programas de aplicación.



60'



Base de datos de ejemplo

López	Briceño	Concepción	c-102	80000
Gonzalez	Zenteno	P. Arenas	c-101	100000
Agular	Balmaceda	Iquique	c-201	180000
			c-305	70000

UADE

## MOLDEANDO EL FUTURO DE LA GESTION Y DEL MANEJO DE LOS DATOS

La evolución de los SGBD se dirige hacia la adaptación a las demandas de datos cada vez mayores y más variadas

- migración hacia entornos de nube
- capacidad de procesamiento en tiempo real
- integración de inteligencia artificial y aprendizaje automático
- el fortalecimiento de la seguridad y privacidad de los datos.

2025



- Independencia física
- Independencia lógica
- Flexibilidad
- Uniformidad
- Sencillez

# Modelo Relacional



## RELACION (TABLA) CLIENTES

### ATRIBUTOS (CABECERA)

ID	CUIT	RAZON SOCIAL	DIRECCION	EMAIL
2152	30225308322	Martinez Herreria	San Martin 445	Martinezh@mh.com
1352	30051255823	Gentiles Ferrum	Montañeses 5001	Adm@gferrum.com
6523	23210004453	STI Soluciones	Congreso 1599	supply@sti.com

### TUPLAS (FILAS)

## CARDINALIDAD

- Relación 1-1
- Relación 1-N o N-1
- Relación N-M

## DOMINIOS

## CLAVES CANDIDATAS

## CLAVES PRIMARIAS

## CLAVES EXTERNAS

## SUPERCLAVE

## MANIPULACION DE DATOS

La manipulación en el modelo relacional se puede realizar mediante:

- **Algebra Relacional:** especifica consultas y manipula datos mediante operaciones como selección, proyección, unión y diferencia, sobre relaciones (tablas). Se centra en la aplicación secuencial de operaciones para obtener resultados deseados, ofreciendo una forma concisa y directa de expresar las instrucciones.
- **Cálculo Relacional:** expresa las consultas en términos de predicados lógicos o fórmulas, especificando las condiciones que deben cumplir las tuplas seleccionadas y centrando la lógica para cumplir con las tuplas de la relación resultante.

## MODELO DE DATOS:

Conjunto de herramientas conceptuales que permiten describir los datos, sus relaciones, límites de integridad que les afectan y terminología a emplear

# Modelo Relacional



## Relación base (varrels base):

Relación autónoma con nombre propio y datos propios.

ID	CUIT	RAZON SOCIAL	DIRECCION	EMAIL
2152	30225308322	Martinez Herreria	San Martin 445	Martinezh@mh.com
1352	30051255823	Gentiles Ferrum	Montañeses 5001	Adm@gferrum.com
6523	23210004453	STI Soluciones	Congreso 1599	supply@sti.com



## Relación derivada:

Relación definida mediante expresiones de álgebra relacional en términos de otras relaciones (Vista) no posee datos propios

SELECT \*  
FROM Clientes  
WHERE Estado='Activo'



## Instantánea:

Relación derivada no actualizable, tiene datos asociados, aunque no propios.

SELECT \*  
FROM Clientes  
WHERE Alta BETWEEN  
'2024-01-01' AND '2024-03-31'



## Resultados de consultas:

Relación derivada generalmente sin nombre, existe mientras existe la sesión en la que fue creada

SELECT \*  
FROM Clientes  
WHERE Alta BETWEEN  
'2024-01-01' AND '2024-03-31'

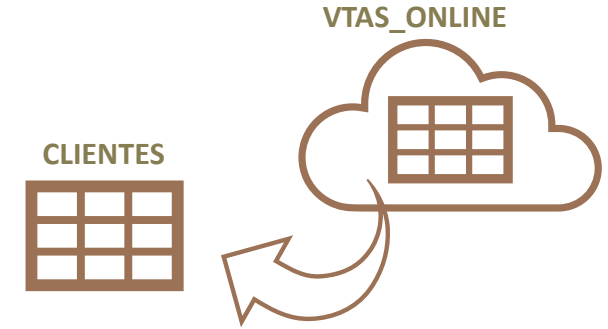


# Modelo Relacional



## Relación Externa (Remote Table):

Involucra datos que se encuentran en una base de datos externa pero que se pueden consultar como si fueran parte de la base de datos local.



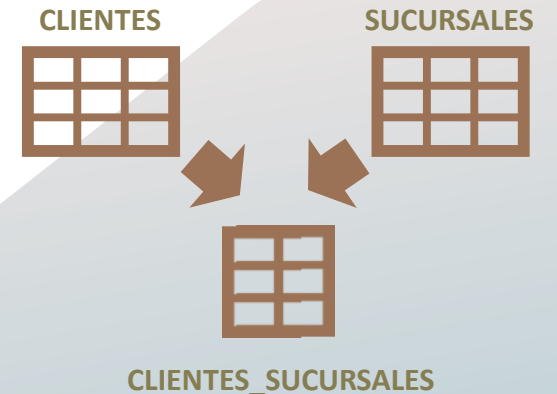
## Relación Materializada (Materialized View):

Similar a una relación derivada, pero los resultados se almacenan físicamente en la base de datos y se actualizan periódicamente.



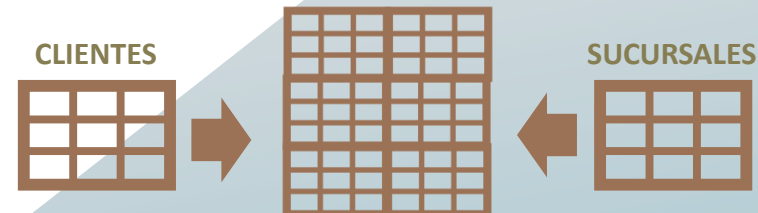
## Relación de Asociación:

Una relación que conecta dos tablas a través de una tabla intermediaria, utilizada para modelar relaciones de muchos a muchos.



## Relación Cruzada (Cross Join):

Una relación temporal que surge del producto cartesiano entre dos tablas, generando todas las combinaciones posibles de filas entre ellas.



# Modelo Relacional

ID	CUIT	RAZON SOCIAL	DIRECCION	EMAIL
2152	30225308322	Martinez Herreria	San Martin 445	Martinezh@mh.com
1352	30051255823	Gentiles Ferrum	Montañeses 5001	Adm@gferrum.com
6523	23210004453	STI Soluciones	Congreso 1599	supply@sti.com



## Relaciones y su Interpretación:

En una base de datos relacional, una relación es una tabla que contiene datos organizados en filas (tuplas) y columnas (atributos). Cada relación tiene un significado específico dentro del contexto del sistema de información. Este significado es una interpretación conceptual de lo que los datos en la tabla representan en la realidad que la base de datos modela.

```
CREATE TABLE CLIENTES (  
  ID INT PRIMARY KEY,  
  CUIT CHAR(11) NOT NULL,  
  RazonSocial VARCHAR(255) NOT NULL,  
  Direccion VARCHAR(255) NOT NULL,  
  Email VARCHAR(255) NOT NULL,  
  CONSTRAINT CK_CUIT_Formato CHECK  
    (CUIT LIKE '##-#####-#')  
);
```



## PREDICADO:

Un predicado es una expresión lógica que, al evaluarse, devuelve verdadero o falso. En el contexto de las bases de datos, un predicado se utiliza para determinar si una tupla (una fila) puede ser parte de una relación.



## PREPOSICION:

Una proposición es un predicado aplicado a una tupla específica. Cuando se inserta o actualiza una fila en una tabla, la proposición se evalúa para determinar si la fila cumple con las reglas de la relación.

```
INSERT INTO CLIENTES (ID, CUIT, RazonSocial, Direccion, Email)  
VALUES (1, '20-12345678-9', 'JPérez S.A.', 'Calle Falsa 123', 'juan.perez@empresa.com');
```

```
INSERT INTO CLIENTES (ID, CUIT, RazonSocial, Direccion, Email)  
VALUES (2, '30123458', 'Gómez SRL', 'Avenida Viva 456', 'contacto@gomez.com');
```



# Modelo Relacional



## Integridad de Entidad

La integridad de entidad asegura que cada relación (tabla) tenga una clave primaria única, garantizando que no existan dos filas en la tabla con la misma **clave primaria**. Esto asegura que cada registro en la tabla sea identificable de manera única.



## Integridad Referencial:

La integridad referencial asegura que las claves foráneas en una tabla apunten siempre a valores válidos en otra tabla. Esto significa que si una tabla tiene una clave foránea que referencia la clave primaria de otra tabla, no se pueden insertar o actualizar valores en la **clave foránea** que no existan en la tabla referenciada.

ID	CUIT	RAZON SOCIAL	DIRECCION	EMAIL
2152	30225308322	Martinez Herreria	San Martin 445	Martinezh@mh.com
1352	30051255823	Gentiles Ferrum	Montañeses 5001	Adm@gferrum.com
6523	23210004453	STI Soluciones	Congreso 1599	supply@sti.com

```
CREATE TABLE CLIENTES (  
  ID INT PRIMARY KEY,  
  CUIT CHAR(11) NOT NULL,  
  RazonSocial VARCHAR(255) NOT NULL,  
  Direccion VARCHAR(255),  
  Email VARCHAR(255)  
);
```

```
CREATE TABLE PEDIDOS (  
  PedidoID INT PRIMARY KEY,  
  ClienteID INT,  
  FechaPedido DATE,  
  Monto DECIMAL(10, 2),  
  FOREIGN KEY (ClienteID) REFERENCES CLIENTES(ID)  
);
```

PEDIDOID	CLIENTEID	FECHAPEDIDO	MONTO
101	2152	2024-01-10	500,00
102	1352	2024-02-15	750,00
103	6523	2024-03-05	250,00



# Modelo Relacional



ID	CUIT	RAZON SOCIAL	DIRECCION	EMAIL
2152	30225308322	Martinez Herreria	San Martin 445	Martinezh@mh.com
1352	30051255823	Gentiles Ferrum	Montañeses 5001	Adm@gferrum.com
6523	23210004453	STI Soluciones	Congreso 1599	supply@sti.com



## Claves Candidatas

Las claves candidatas son subconjuntos de atributos en una tabla que podrían actuar como una clave primaria porque cumplen con las propiedades de unicidad y minimalidad. Cada clave candidata identifica de manera única una fila en la tabla, y no se puede eliminar ningún atributo de la clave sin perder esta propiedad.



## Clave Primaria:

De todas las claves candidatas, se selecciona una para ser la clave primaria. La clave primaria identifica de manera única cada fila en la tabla y no puede contener valores nulos.



## Claves Simples:

Una clave que consiste en un solo atributo.



## Claves Compuestas:

Una clave que consiste en dos o más atributos.  
La combinación de estos atributos es única para cada fila.



## Claves Foráneas

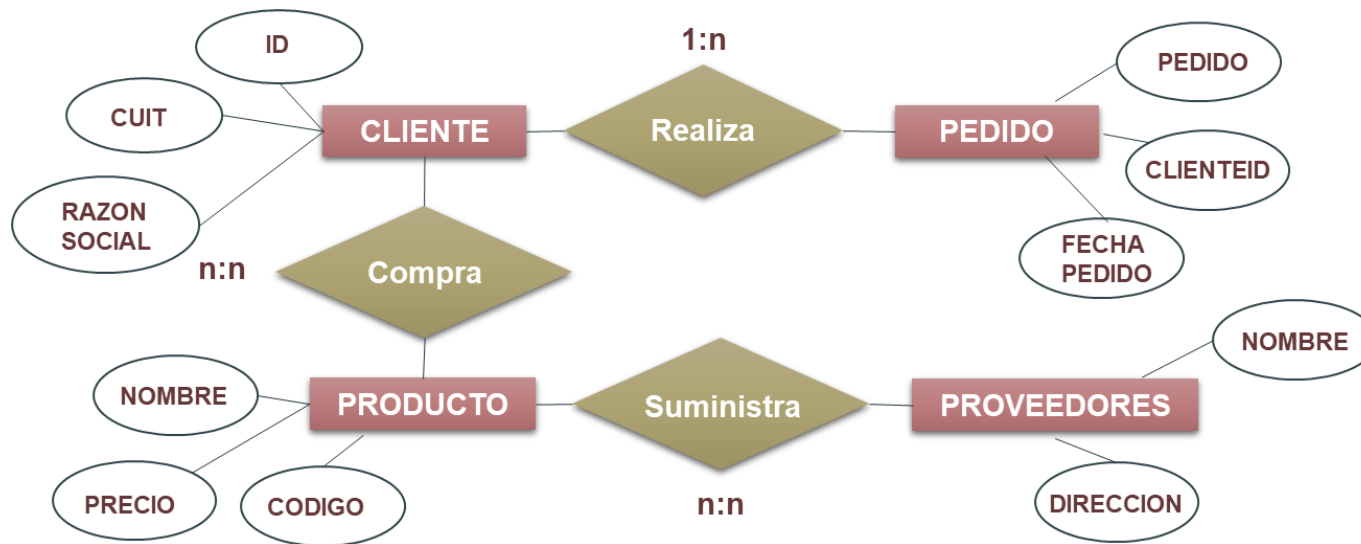
Una clave foránea es un atributo en una tabla que hace referencia a la clave primaria de otra tabla. Las claves foráneas permiten establecer y reforzar las relaciones entre diferentes tablas.

# Modelo Relacional



## Diagrama de Entidad-Relación (ERD)

Es una representación gráfica que muestra las entidades de un sistema de base de datos y las relaciones entre ellas. Los ERD se utilizan principalmente durante la fase de diseño conceptual para modelar la estructura de los datos y entender cómo interactúan las diferentes entidades en un sistema.



## Cardinalidad

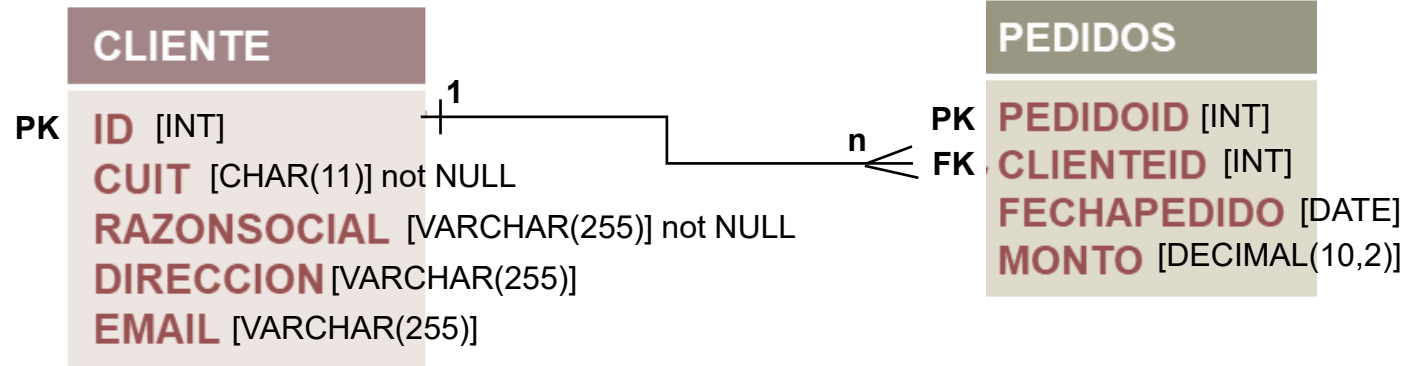
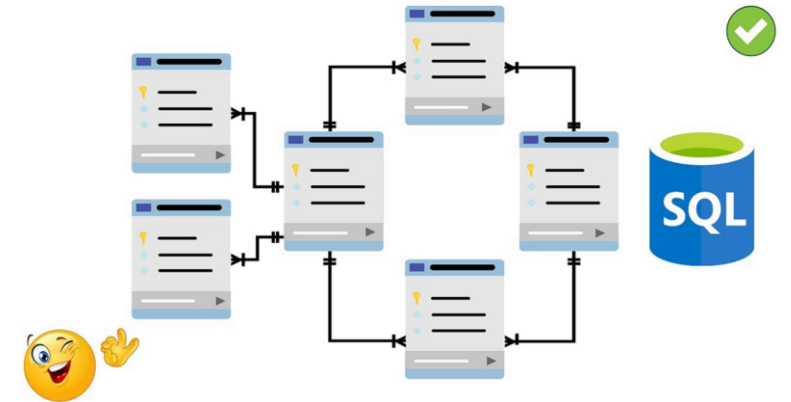
Define el número de instancias de una entidad que pueden estar asociadas con las instancias de otra entidad.  
**1:1, 1:n, n:1, n:n**

# Modelo Relacional



## Diagramas de Base de Datos

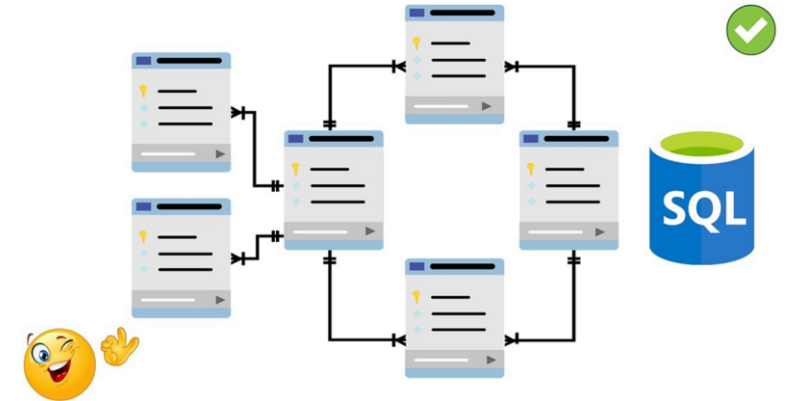
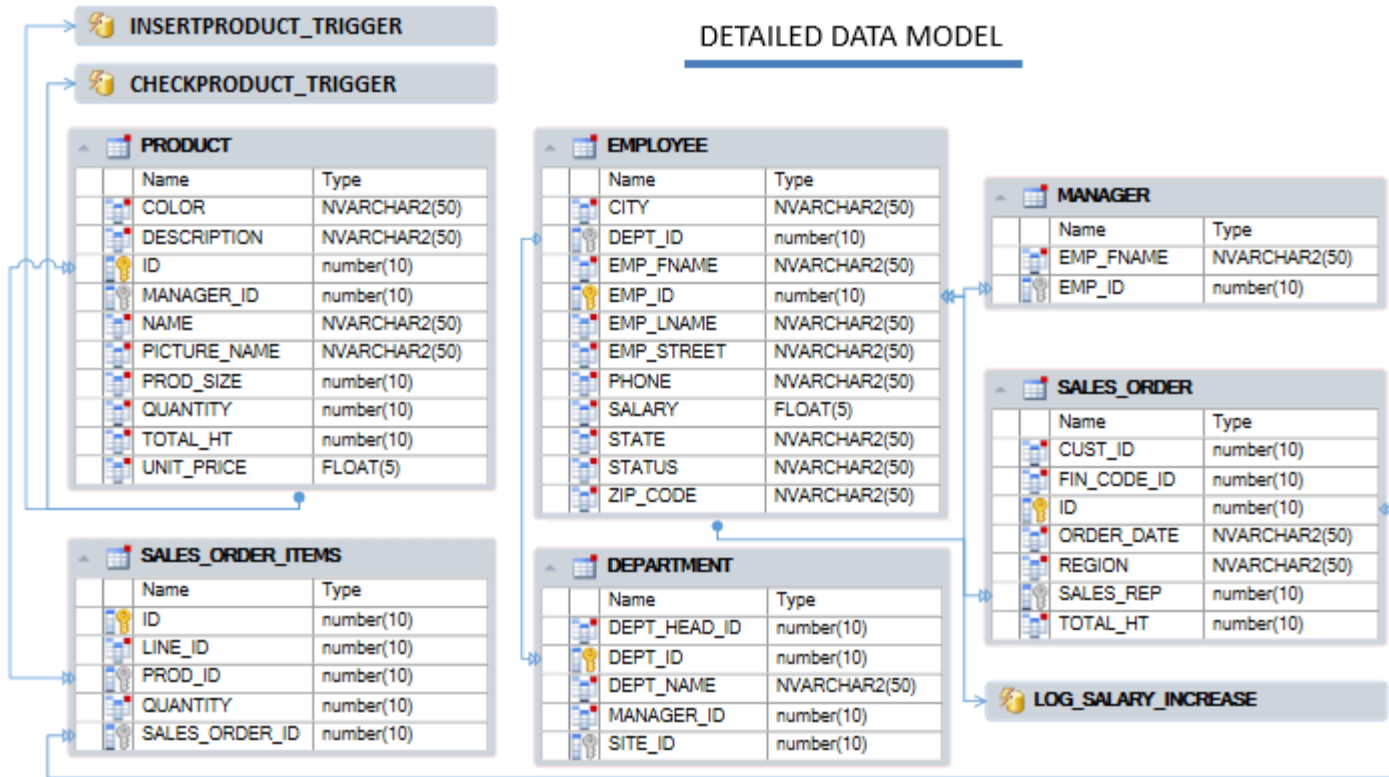
Es una representación técnica y detallada de cómo las entidades y relaciones se implementan en la base de datos física. Este diagrama incluye información específica sobre los tipos de datos, claves primarias, claves foráneas, índices, y otros aspectos técnicos relacionados con la implementación en un sistema de gestión de bases de datos (DBMS) como SQL Server.



# Modelo Relacional



## Diagramas de Base de Datos

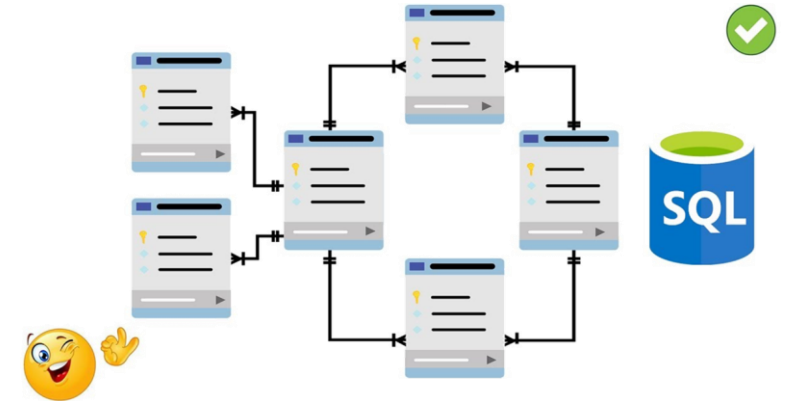
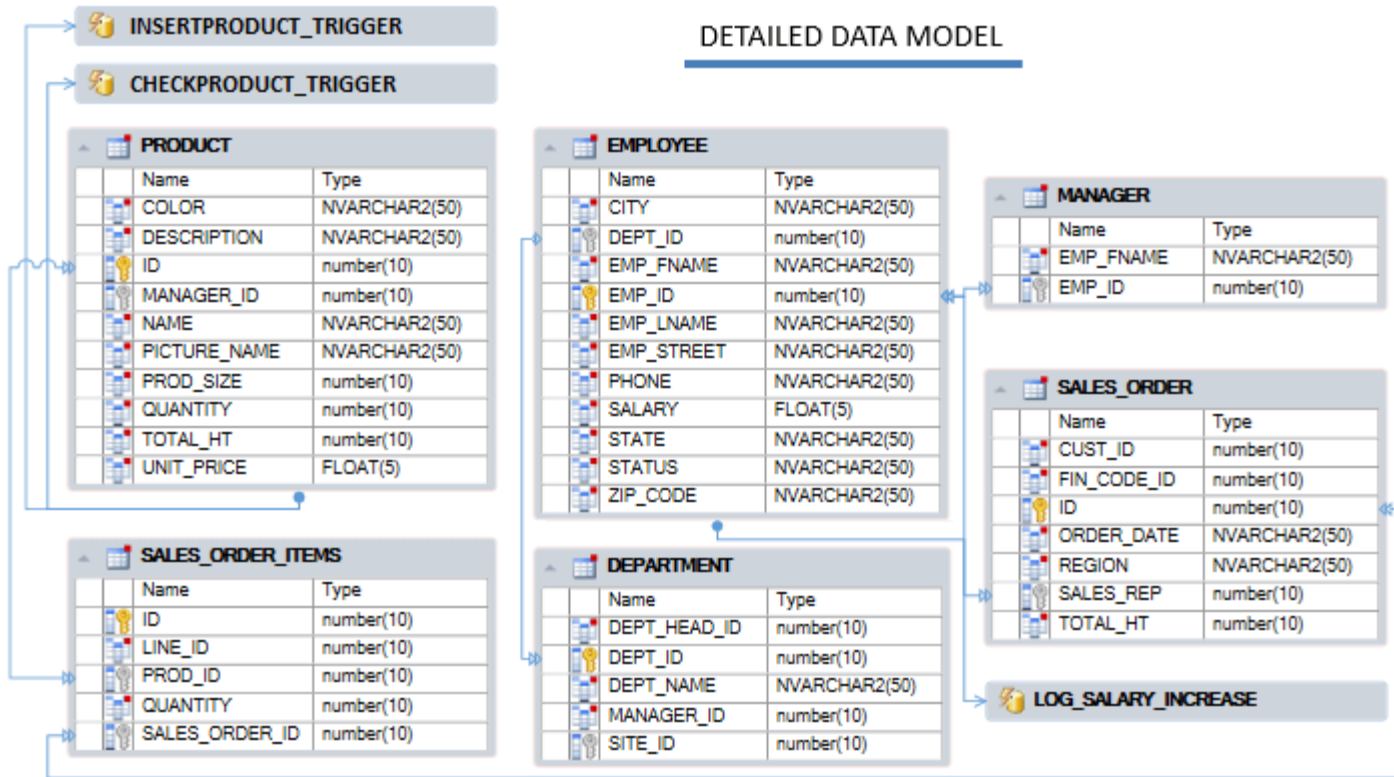




# Modelo Relacional



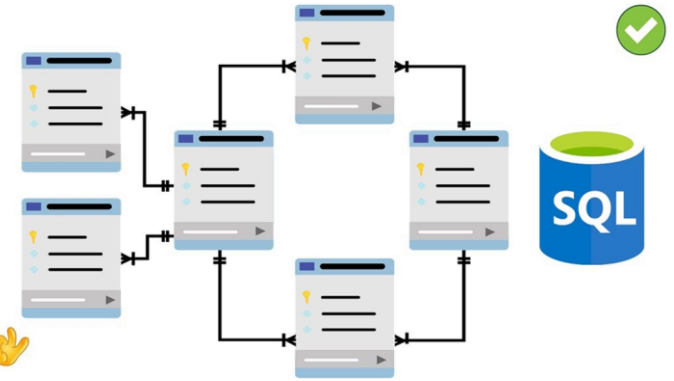
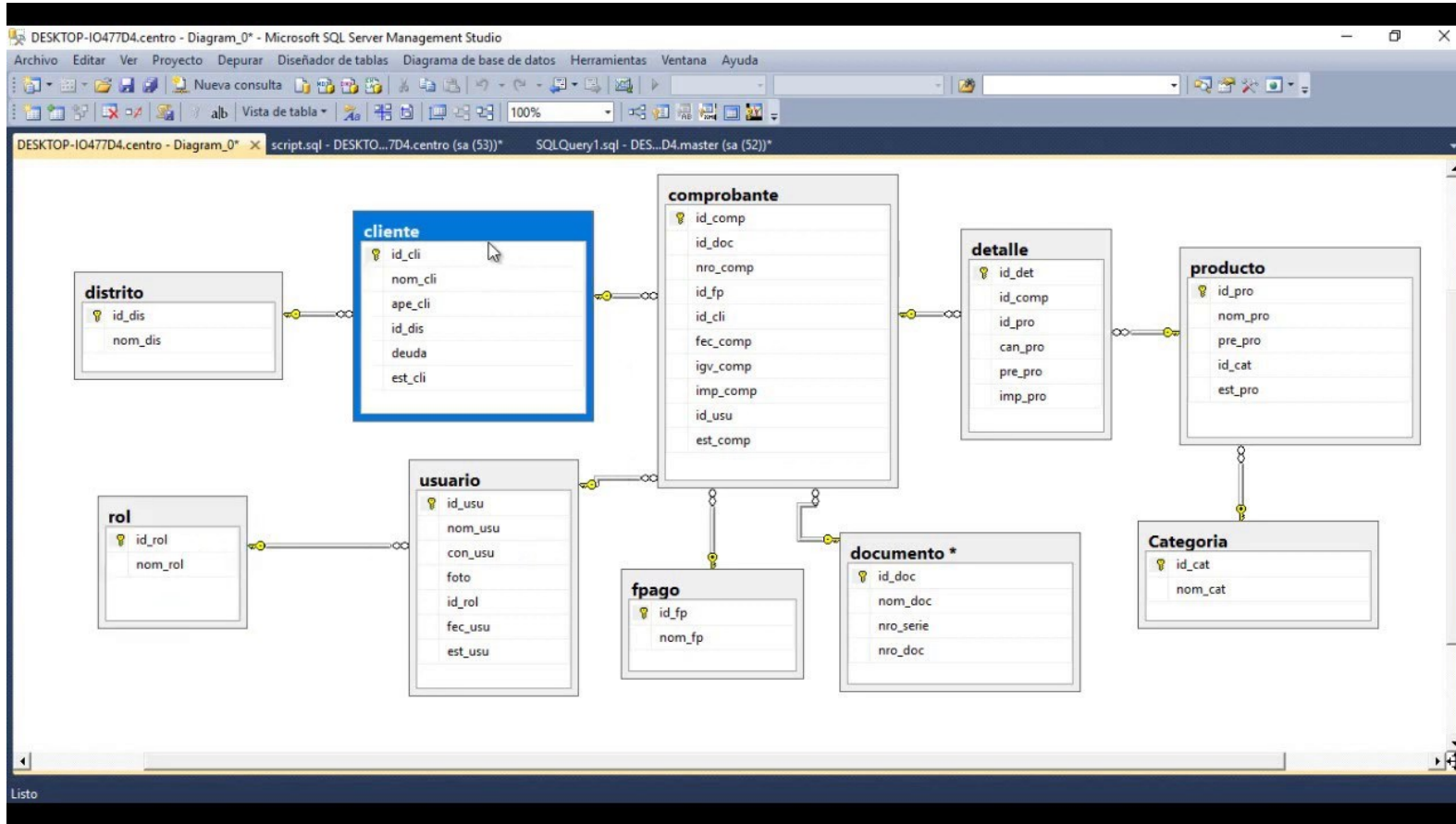
## Diagramas de Base de Datos



# Modelo Relacional

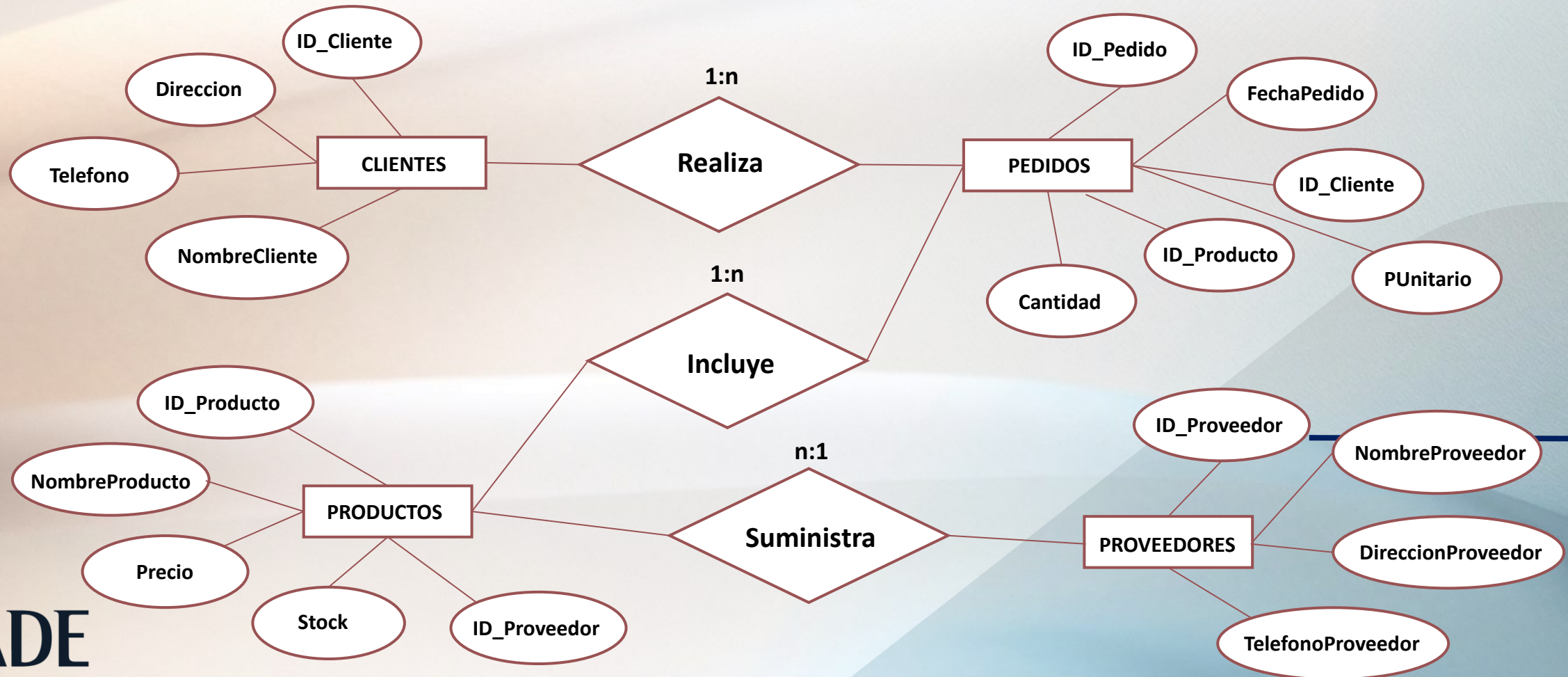


## Diagramas de Base de Datos



# Creación de Tablas y Definición de Claves

A partir del Diagrama de Entidad-Relación (ERD) dado, crear la base de datos **GestionPedidos** y las tablas **Cliente**, **Pedido**, **Producto** y **Proveedores** con sus respectivas claves primarias y foráneas.



# Creación de Tablas y Definición de Claves



Creación de la Base de Datos: comando CREATE DATABASE

```
CREATE DATABASE GestionPedidos;
```

vez creada la base de datos, necesitas seleccionarla para asegurarte de que las tablas se creen dentro de ella, comando USE

```
USE GestionPedidos;
```

Creación de Tablas y Definición de Claves

```
CREATE TABLE Clientes (  
    ID_Cliente INT PRIMARY KEY,  
    NombreCliente VARCHAR(100),  
    Direccion VARCHAR(150),  
    Telefono VARCHAR(15)  
);
```



# Diagramas de Base de Datos



Pasos para Crear un Diagrama de Base de Datos en SQL Server Management Studio (SSMS)

Abrir SQL Server Management Studio (SSMS)

Seleccionar la Base de Datos

Haz clic derecho sobre la base de datos GestionPedidos.

Selecciona Diagramas de Base de Datos y luego haz clic en Nuevo Diagrama de Base de Datos.

Agregar Tablas al Diagrama:

Aparecerá una ventana que muestra todas las tablas disponibles en la base de datos.

Selecciona las tablas Clientes, Proveedores, Productos, y Pedidos, y haz clic en Agregar.

Para guardar, haz clic en Guardar en la barra de herramientas o selecciona Archivo > Guardar en el menú.



# Consultas Básicas: SELECT, FROM y WHERE



## SELECT

- El comando SELECT se utiliza para especificar las columnas que se desea recuperar de una o más tablas en una base de datos.
- Es el comando principal en SQL para extraer datos.
- Puedes seleccionar una o varias columnas especificando sus nombres.
- También puedes usar SELECT \* para seleccionar todas las columnas de la tabla.
- Permite realizar operaciones sobre los datos, como concatenación de columnas, funciones de agregación (SUM, AVG), y más.

```
SELECT nombre, edad FROM empleados;
```

# Consultas Básicas: SELECT, FROM y WHERE



## FROM

- El comando FROM se utiliza para especificar la tabla o tablas de las que se van a recuperar los datos en una consulta SQL.
- Después de FROM, se indica el nombre de la tabla de la cual se van a extraer los datos.
- Si se necesitan datos de varias tablas, se pueden unir estas tablas mediante la cláusula JOIN.
- También puedes usar alias para tablas para simplificar la consulta.

```
SELECT * FROM empleados;
```



# Consultas Básicas: SELECT, FROM y WHERE



## WHERE

- El comando WHERE se utiliza para especificar condiciones que los datos deben cumplir para ser seleccionados.
- Se emplea para filtrar los resultados de una consulta SQL.
- Solo las filas que cumplen las condiciones especificadas en WHERE serán incluidas en el resultado.
- Puedes combinar múltiples condiciones usando operadores lógicos como AND, OR, y NOT.
- También permite el uso de operadores de comparación como =, !=, <, >, <=, >=, entre otros.

```
SELECT nombre FROM empleados WHERE edad > 30;
```



# Prácticas

---



## Práctica 1

- Obtener una lista de todos los productos cuyo precio sea superior a \$20.

## Práctica 2

- Listar todos los pedidos que se realizaron el 3 de agosto de 2024.

## Práctica 3

- Obtener una lista de todos los clientes que viven en "Calle Falsa 123".

# Modelo relacional

---



El modelo relacional es crucial en la manipulación de datos para optimizar consultas, garantizar la precisión y fiabilidad del análisis, y estandarizar operaciones en bases de datos relacionales.

Proporciona una base teórica sólida para lenguajes declarativos como SQL, facilitando el acceso y análisis eficiente de grandes volúmenes de datos.

Su aplicación permite a las organizaciones mejorar el rendimiento de las consultas y obtener insights valiosos de manera rápida y confiable, potenciando la toma de decisiones estratégicas basadas en datos.

# Modelo relacional

---



## Bibliografía:

Introducción a los Sistemas de Bases de Datos - C. J. Date – 7º Edición

- **CAPÍTULO 3: Una introducción a las bases de datos relacionales.**
- **CAPÍTULO 5: Dominios, relaciones y varrels base.**

Sistemas de Bases de Datos – Conceptos Fundamentales - Elmasri / Navathe – 5º Edición.

- **Capítulo 5: El modelo de datos relacional y las restricciones de una base de datos relacional.**
- **Capítulo 6: El álgebra relacional y los cálculos relacionales.**



## Próximo encuentro

---



### *Módulo 2: Creación de Estructuras y Álgebra Relacional*

Álgebra Relacional: introducción y principales operaciones.  
Operadores básicos: selección, proyección, unión, diferencia.