**Informe sobre Bases de Datos Relacionales**

**1. Las 12 Reglas de Codd**

Edgar F. Codd, el creador del modelo relacional, propuso 12 reglas para definir los requisitos que un sistema de gestión de bases de datos debe cumplir para ser considerado verdaderamente relacional. Estas reglas son:

1. **Regla Fundamental**, establece que un sistema de gestión de bases de datos debe ser relacional para ser considerado como tal. Es decir, debe utilizar sus capacidades relacionales exclusivamente para gestionar la base de datos. Esta regla es la base sobre la cual se construyen las otras 12 reglas de Codd.
2. **Regla de la Información:** Toda la información en una base de datos relacional se representa de forma explícita en el nivel lógico y exactamente de una manera: con valores en tablas. Esto significa que todos los datos deben estar organizados en tablas, y cada tabla debe tener un nombre único.
3. **Regla del Acceso Garantizado:** Se garantiza que todos y cada uno de los datos (valor atómico) de una base de datos relacional son accesibles lógicamente mediante una combinación de nombre de tabla, valor de clave primaria y nombre de columna. Esto asegura que cada dato sea accesible sin ambigüedad.
4. **Regla del Tratamiento Sistemático de Valores Nulos:** Los sistemas de gestión de base de datos plenamente relacionales admiten los valores nulos (distintos de la cadena vacía, los blancos, los ceros o cualquier otro número) para representar la información desconocida y la inaplicable de manera sistemática e independiente del tipo de dato. Esto permite manejar datos incompletos o desconocidos de manera uniforme.
5. **Catálogo Dinámico en Línea Basado en el Modelo Relacional:** La descripción de la base de datos se representa a nivel lógico igual que los datos comunes, de modo que los usuarios autorizados pueden utilizar el mismo lenguaje relacional en su consulta que el que aplican a los datos comunes. Esto implica que la estructura de la base de datos debe ser accesible y manipulable mediante el mismo lenguaje que se usa para los datos.
6. **Regla del Sublenguaje de Datos Completo:** Un sistema relacional debe permitir varios lenguajes y varios modos de uso terminal (como rellenar formularios, por ejemplo). Sin embargo, debe haber al menos un lenguaje cuyas declaraciones se puedan expresar, mediante una sintaxis bien definida, como cadenas de caracteres y que respalde de forma integral los siguientes aspectos: definición de datos, definición de vistas, manipulación de datos (interactiva y por programa), restricciones de integridad y límites de transacción (begin, commit y rollback).
7. **Regla de Actualización de Vistas**: Todas las vistas que son teóricamente actualizables son también actualizables por el sistema. Esto significa que si una vista puede ser actualizada en teoría, el sistema debe permitir su actualización en la práctica.
8. **Inserción, Actualización y Borrado de Alto Nivel:** La capacidad de gestionar una relación base o una relación derivada como un solo operando no solo se aplica a la recuperación de los datos, sino también a la inserción, actualización y eliminación de datos. Esto implica que las operaciones deben poder realizarse sobre conjuntos de datos, no solo sobre registros individuales.
9. **Independencia Física de los Datos**: Los programas de aplicación y actividades terminales permanecen inalterados a nivel lógico cuando se realizan cambios en las representaciones de almacenamiento o en los métodos de acceso. Esto asegura que los cambios en la forma en que los datos se almacenan físicamente no afecten la lógica de las aplicaciones.
10. **Independencia Lógica de los Datos:** Los programas de aplicación y actividades terminales permanecen inalterados a nivel lógico cuando se realizan cambios en las tablas base que preservan la información. La independencia de datos lógica es más difícil de lograr que la independencia física de datos, pero es crucial para la flexibilidad del sistema.
11. **Independencia de la Integridad:** Las restricciones de integridad específicas para una determinada base de datos relacional se deben poder definir en el sublenguaje de datos relacional y almacenar en el catálogo, no en los programas de aplicación. Esto permite que las restricciones de integridad se gestionen de manera centralizada y no se mezclen con la lógica de la aplicación.
12. **Independencia de la Distribución:** El usuario final no ha de ver que los datos están distribuidos en varias ubicaciones. Los usuarios deben tener siempre la impresión de que los datos se encuentran en un solo lugar, independientemente de su distribución física.
13. **Regla de la No Subversión:** Si un sistema relacional tiene un lenguaje de bajo nivel (un registro cada vez), ese nivel bajo no puede utilizarse para subvertir o eludir las reglas y restricciones de integridad expresadas en el lenguaje relacional de alto nivel (varios registros cada vez). Esto asegura que las operaciones de bajo nivel no comprometan la integridad del sistema.

**2. Normalización de Base de Datos**

La normalización es el proceso de organizar los datos en una base de datos para reducir la redundancia y mejorar la integridad de los datos. Las formas normales más comunes son:

* **Primera Forma Normal (1NF):**

La Primera Forma Normal (1NF) establece que una tabla debe cumplir con los siguientes criterios:

1. **Atomicidad**: Cada columna debe contener valores atómicos, es decir, indivisibles.
2. **Clave Primaria**: Cada tabla debe tener una clave primaria que identifique de manera única cada fila.
3. **No Repetición**: No debe haber filas duplicadas ni columnas repetidas.

**Ejemplo**: Imaginemos una tabla de clientes:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

Para cumplir con 1NF, debemos separar los teléfonos en filas individuales:

Texto

Descripción generada automáticamente con confianza media

* **Segunda Forma Normal (2NF)**

La Segunda Forma Normal (2NF) se basa en 1NF y añade los siguientes criterios:

**Dependencia Completa de la Clave Primaria:** Todos los atributos no clave deben depender completamente de la clave primaria.

Ejemplo: Continuando con el ejemplo anterior, supongamos que tenemos una tabla de pedidos:

Interfaz de usuario gráfica, Texto, Aplicación

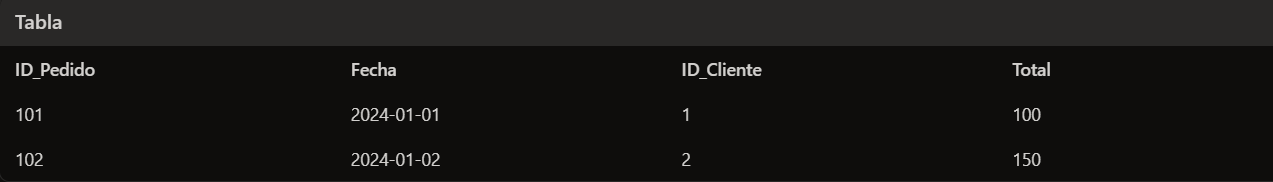
Descripción generada automáticamente

Para cumplir con 2NF, debemos eliminar la redundancia de los datos del cliente:

**Tabla Clientes**:



**Tabla Pedidos**:



* **Tercera Forma Normal (3NF)**

La Tercera Forma Normal (3NF) se basa en 2NF y añade el siguiente criterio:

1. **Eliminación de Dependencias Transitivas:** Los atributos no clave no deben depender de otros atributos no clave.

**Ejemplo:** Supongamos que tenemos una tabla de productos:

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

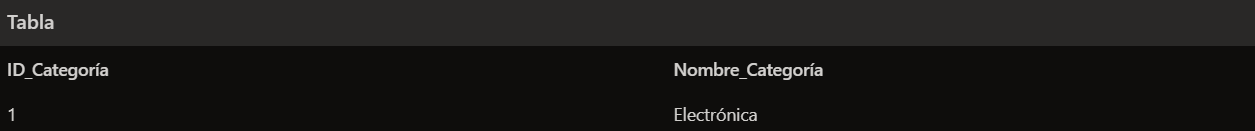
Para cumplir con 3NF, debemos eliminar la dependencia entre Categoría y Precio:

**Tabla Productos:**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Tabla Categorías:**

****

**3. El Modelo Relacional para el Modelado y la Gestión de Bases de Datos**

El modelo relacional, propuesto por Edgar F. Codd en 1970, es un enfoque para la gestión de bases de datos que organiza los datos en tablas (relaciones) compuestas por filas (tuplas) y columnas (atributos). Este modelo se basa en la teoría de conjuntos y la lógica de predicados, y es el más utilizado debido a su simplicidad y flexibilidad.

**Características del Modelo Relacional**

1. **Estructura de Datos**:
   * **Tablas**: Los datos se almacenan en tablas bidimensionales.
   * **Tuplas**: Cada fila de una tabla representa una tupla, que es un conjunto de valores relacionados.
   * **Atributos**: Cada columna de una tabla representa un atributo, que describe una característica de los datos.
2. **Manipulación de Datos**:
   * **Operaciones Relacionales**: Se utilizan operaciones como selección, proyección, unión, intersección y diferencia para manipular los datos.
   * **Lenguaje de Consulta**: SQL (Structured Query Language) es el lenguaje estándar para interactuar con bases de datos relacionales.
3. **Integridad de Datos**:
   * **Integridad de Entidad**: Cada tabla debe tener una clave primaria que identifique de manera única cada tupla.
   * **Integridad Referencial**: Las claves externas deben corresponder a claves primarias en otras tablas, asegurando la consistencia de las relaciones entre tablas.
   * **Integridad de Dominio**: Los valores de los atributos deben pertenecer a un dominio específico, es decir, un conjunto de valores permitidos.

**Ventajas del Modelo Relacional**

1. **Simplicidad**: El uso de tablas facilita la comprensión y el manejo de los datos.
2. **Flexibilidad**: Permite realizar consultas complejas y obtener información detallada mediante el uso de SQL.
3. **Integridad de Datos**: Las restricciones de integridad aseguran la consistencia y precisión de los datos.
4. **Escalabilidad**: Es adecuado para manejar grandes volúmenes de datos y puede escalar según las necesidades de la organización.