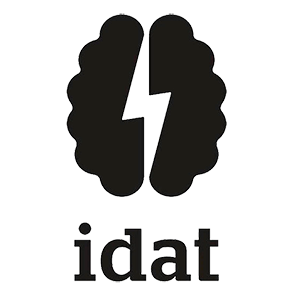
****

**BASE DE DATOS RELACIONALES**

**Programa de Especialización en Desarrollo Back-end**

**PROFESOR: rOBERTO PINEDA**

**aLUMNO: SILVA URIARTE ELMER**

**lima-peru**

**2024**

**INTRODUCCION**

Las bases de datos relacionales son un tipo de sistema de gestión de bases de datos (SGBD) que organiza los datos en tablas estructuradas. Cada tabla, conocida como relación, almacena datos en filas y columnas, donde cada fila representa un registro único y cada columna un atributo del registro. Este enfoque facilita la organización, el acceso y la manipulación eficiente de grandes cantidades de datos.

**Las Reglas de Codd**

1. **Regla de la Información**
   * **Definición**: Todos los datos se representan en tablas, y toda la información está almacenada en tablas.
   * **Ejemplo**: Una tabla Clientes con columnas ID, Nombre, y Correo.
2. **Regla del Acceso Garantizado**
   * **Definición**: Cada dato debe ser accesible por combinación de nombre de tabla, clave primaria y nombre de columna.
   * **Ejemplo**: Para acceder al correo de un cliente, se usa SELECT Correo FROM Clientes WHERE ID = 1;.
3. **Tratamiento Sistemático de Valores Nulos**
   * **Definición**: Los valores nulos deben ser soportados y manejados de manera uniforme.
   * **Ejemplo**: El campo Telefono puede ser nulo si un cliente no ha proporcionado un número.
4. **Catálogo en Línea Basado en el Modelo Relacional**
   * **Definición**: El sistema debe tener un catálogo accesible mediante el lenguaje de consultas.
   * **Ejemplo**: Usar INFORMATION\_SCHEMA para obtener metadatos de las tablas.
5. **Sub-lenguaje de Datos Completo**
   * **Definición**: El sistema debe soportar un lenguaje de datos completo, como SQL, para definición, manipulación y control de datos.
   * **Ejemplo**: Crear una tabla, insertar datos, y realizar consultas con SQL.
6. **Actualización de Vistas**
   * **Definición**: Las vistas deben ser actualizables.
   * **Ejemplo**: Crear una vista de ClientesActivos y actualizar el correo de un cliente en esta vista.
7. **Inserción, Actualización y Borrado de Alto Nivel**
   * **Definición**: Debe ser posible realizar estas operaciones en conjuntos de datos completos.
   * **Ejemplo**: Aumentar el salario de todos los empleados de un departamento con una sola consulta SQL.
8. **Independencia Física de Datos**
   * **Definición**: Los cambios en la estructura física no deben afectar la estructura lógica.
   * **Ejemplo**: Cambiar la ubicación de los archivos de datos sin modificar las consultas SQL.
9. **Independencia Lógica de Datos**
   * **Definición**: Los cambios en la estructura lógica no deben afectar las aplicaciones que usan los datos.
   * **Ejemplo**: Añadir una columna Edad a la tabla Empleados sin alterar las consultas existentes.
10. **Integridad de los Datos**
    * **Definición**: Deben existir restricciones de integridad específicas del usuario.
    * **Ejemplo**: Definir claves primarias y foráneas para asegurar la consistencia de los datos.
11. **Independencia de Distribución**
    * **Definición**: Los usuarios no deben necesitar saber si los datos están distribuidos en diferentes ubicaciones.
    * **Ejemplo**: Consultar una base de datos distribuida sin conocer la distribución física.
12. **Regla de No Subversión**
    * **Definición**: Ningún nivel inferior debe permitir eludir las reglas relacionales.
    * **Ejemplo**: Asegurar que todos los accesos a datos pasen por el sistema de gestión de bases de datos (SGBD).
13. **Línea de Tiempo**

|  |  |
| --- | --- |
| Año | Evento Clave |
| 1970 | Edgar F. Codd publica el artículo "A Relational Model of Data for Large Shared Data Banks", introduciendo el concepto de bases de datos relacionales. |
| 1974 | Desarrollo del lenguaje de consulta SEQUEL por IBM, precursor de SQL, para trabajar con bases de datos relacionales. |
| 1979 | Oracle lanza el primer SGBD relacional comercial basado en SQL. |
| 1981 | Edgar F. Codd recibe el Premio Turing por sus contribuciones al modelo relacional. |
| 1985 | **Codd formula las 12 reglas para definir un SGBD relacional.** |
| 1986 | ANSI e ISO estandarizan SQL como el lenguaje estándar para la gestión de bases de datos relacionales. |
| 1987 | IBM lanza DB2, un SGBD relacional comercial, que implementa muchas de las reglas de Codd. |
| 1989 | Publicación de las normas SQL-89 por ANSI, incluyendo reglas de integridad referencial y procedimientos almacenados. |
| 1992 | SQL-92 se convierte en el estándar internacional, mejorando el soporte para las reglas de Codd en los SGBD comerciales. |
| 1995 | MySQL se lanza como un SGBD relacional de código abierto, adoptando muchas de las reglas de Codd. |
| 2003 | SQL:2003 introduce características avanzadas, como XML y OLAP, alineándose con las reglas de Codd en cuanto a robustez y funcionalidad. |
| 2008 | SQL:2008 agrega soporte para nuevas funciones y tipos de datos, mejorando la conformidad con las reglas de Codd. |
| 2011 | SQL:2011 introduce soporte para procesamiento de datos temporal y más funciones analíticas, continuando con la mejora en la implementación de las reglas. |
| 2016 | SQL:2016 amplía el soporte para JSON, fortaleciendo las capacidades relacionales y la conformidad con las reglas de Codd. |
| 2021 | Los SGBD modernos como PostgreSQL, Oracle, MySQL, y SQL Server siguen evolucionando, implementando y mejorando el soporte para las 12 reglas de Codd. |

**Conclusión**

Las 12 reglas de Codd proporcionan un marco conceptual para evaluar y diseñar sistemas de bases de datos relacionales. Aunque algunos SGBD comerciales pueden no cumplir con todas estas reglas en su totalidad, siguen siendo una guía valiosa para garantizar que las bases de datos sean eficientes, coherentes y accesibles. Cumplir con estas reglas asegura que los sistemas de bases de datos mantengan su integridad y funcionen de manera óptima.

**Normalización de Base de Datos**

La normalización de bases de datos es un proceso fundamental en el diseño de bases de datos relacionales. Su objetivo es organizar los datos para minimizar la redundancia y evitar problemas de integridad. Este proceso implica descomponer tablas grandes en tablas más pequeñas y definir relaciones entre ellas. Este informe describe las distintas formas normales y su importancia en la normalización de bases de datos.

**Formas Normales**

1. **Primera Forma Normal (1NF)**
   * **Definición**: Una tabla está en 1NF si todos sus atributos contienen valores atómicos y cada columna contiene solo un valor de un solo tipo de dato.
   * **Ejemplo**:
     + Tabla no normalizada:

|  |  |  |
| --- | --- | --- |
| ID | Nombre | Teléfonos |
| 1 | Juan | 123-456, 789-012 |

* + - Tabla en 1NF:

|  |  |  |
| --- | --- | --- |
| ID | Nombre | Teléfono |
| 1 | Juan | 123-456 |
| 1 | Juan | 789-012 |

1. **Segunda Forma Normal (2NF)**
   * **Definición**: Una tabla está en 2NF si está en 1NF y todos los atributos no clave son dependientes de la clave primaria completa.
   * **Ejemplo**:
     + Tabla en 1NF:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Nombre | Curso | Profesor |
| 1 | Juan | Matemáticas | Pérez |

* + - Descomposición en 2NF:
      * Tabla Estudiantes:

|  |  |
| --- | --- |
| ID | Nombre |
| 1 | Juan |

* + - * Tabla Cursos:

|  |  |
| --- | --- |
| Curso | Profesor |
| Matemáticas | Pérez |

1. **Tercera Forma Normal (3NF)**
   * **Definición**: Una tabla está en 3NF si está en 2NF y no contiene dependencias transitivas, es decir, los atributos no clave deben depender únicamente de la clave primaria.
   * **Ejemplo**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Nombre | Curso | Profesor | Departamento |
| 1 | Juan | Matemáticas | Pérez | Ciencias |

* + - Tabla en 2NF:
    - Descomposición en 3NF:
      * Tabla Estudiantes:

|  |  |
| --- | --- |
| ID | Nombre |
| 1 | Juan |

* + - * Tabla Cursos:

|  |  |  |
| --- | --- | --- |
| Curso | Profesor | DeptID |
| Matemáticas | Pérez | 1 |

* + - * Tabla Departamentos:

|  |  |
| --- | --- |
| DeptID | Departamento |
| 1 | Ciencias |

1. **Forma Normal de Boyce-Codd (BCNF)**
   * **Definición**: Una tabla está en BCNF si está en 3NF y para cada dependencia funcional X → Y, X es una superclave.
   * **Ejemplo**:
     + Tabla en 3NF:

|  |  |  |
| --- | --- | --- |
| Curso | Profesor | Aula |
| Matemáticas | Pérez | A101 |

* + - Descomposición en BCNF:
      * Tabla Cursos:

|  |  |
| --- | --- |
| Curso | Profesor |
| Matemáticas | Pérez |

* + - * Tabla Aulas:

|  |  |
| --- | --- |
| Curso | Aula |
| Matemáticas | A101 |

1. **Cuarta Forma Normal (4NF)**
   * **Definición**: Una tabla está en 4NF si está en BCNF y no tiene dependencias multivaluadas.
   * **Ejemplo**:
     + Tabla con dependencias multivaluadas:

|  |  |  |
| --- | --- | --- |
| Curso | Profesor | Aula |
| Matemáticas | Pérez | A101 |
| Matemáticas | Pérez | A102 |

* + - Descomposición en 4NF:
      * Tabla Cursos:

|  |  |
| --- | --- |
| Curso | Profesor |
| Matemáticas | Pérez |

* + - * Tabla CursoAulas:

|  |  |
| --- | --- |
| Curso | Aula |
| Matemáticas | A101 |
| Matemáticas | A102 |

1. **Quinta Forma Normal (5NF)**
   * **Definición**: Una tabla está en 5NF si está en 4NF y no contiene dependencias de unión.
   * **Ejemplo**:
     + Tabla en 4NF:

|  |  |  |
| --- | --- | --- |
| Curso | Profesor | Aula |
| Matemáticas | Pérez | A101 |

* + - Descomposición en 5NF:
      * Tabla Cursos:

|  |  |
| --- | --- |
| Curso | Profesor |
| Matemáticas | Pérez |

* + - * Tabla CursoAulas:

|  |  |
| --- | --- |
| Curso | Aula |
| Matemáticas | A101 |

**Importancia de la Normalización**

La normalización de bases de datos es crucial para:

* **Minimizar la Redundancia**: Evitar la duplicación de datos, lo que reduce el espacio de almacenamiento y mejora la eficiencia.
* **Mejorar la Integridad de los Datos**: Asegurar que los datos sean consistentes y correctos mediante la eliminación de anomalías de inserción, actualización y borrado.
* **Facilitar el Mantenimiento**: Hacer que la base de datos sea más fácil de mantener y actualizar, ya que las modificaciones se realizan en un solo lugar.

|  |  |
| --- | --- |
| Año | Evento Clave |
| 1970 | Edgar F. Codd publica el artículo sobre el modelo relacional. |
| 1971 | Codd introduce las primeras tres formas normales (1NF, 2NF, 3NF). |
| 1974 | Boyce-Codd desarrolla la Forma Normal de Boyce-Codd (BCNF). |
| 1977 | Introducción de la Cuarta Forma Normal (4NF) por Ronald Fagin. |
| 1979 | Introducción de la Quinta Forma Normal (5NF) por Ronald Fagin. |
| 1981 | Desarrollo de técnicas para la implementación de la normalización en SGBD comerciales. |
| 1986 | ANSI estandariza SQL, que facilita la implementación de formas normales. |
| 1990 | Se mejora el entendimiento y la aplicación de formas normales en bases de datos distribuidas. |

**Conclusión**

La normalización es un proceso esencial en el diseño de bases de datos relacionales. A través de las diversas formas normales, se logra una estructura de datos eficiente, sin redundancias y con integridad garantizada. La implementación de la normalización permite a las organizaciones manejar sus datos de manera más efectiva y confiable.

#### Modelo Relacional para el Modelado y la Gestión de Bases de Datos

El modelo relacional es un paradigma para la gestión de bases de datos que organiza los datos en tablas (o relaciones) de forma estructurada y eficiente. Propuesto por Edgar F. Codd en 1970, el modelo relacional ha transformado la forma en que se almacenan, gestionan y recuperan los datos, siendo la base de la mayoría de los sistemas de gestión de bases de datos (SGBD) actuales.

#### Principios del Modelo Relacional

1. **Tablas como Estructuras Fundamentales**
   * En el modelo relacional, los datos se organizan en tablas. Cada tabla tiene un nombre único y se compone de filas (tuplas) y columnas (atributos). Las tablas representan entidades del mundo real y las relaciones entre ellas.
2. **Atributos y Dominios**
   * Cada columna en una tabla representa un atributo y tiene un dominio, que es el conjunto de valores posibles que puede tomar. Por ejemplo, una columna Edad puede tener un dominio de valores enteros positivos.
3. **Claves Primarias**
   * Cada tabla debe tener una clave primaria, un conjunto de uno o más atributos que identifican de manera única cada fila en la tabla. Las claves primarias garantizan que no haya filas duplicadas y facilitan la recuperación de datos.
4. **Claves Foráneas**
   * Las claves foráneas se utilizan para establecer relaciones entre tablas. Una clave foránea en una tabla es una referencia a la clave primaria de otra tabla, creando vínculos que representan las relaciones entre entidades.
5. **Integridad de Datos**
   * El modelo relacional incluye reglas de integridad para asegurar la consistencia y validez de los datos. Estas reglas incluyen la integridad de entidad (la clave primaria no puede ser nula), la integridad referencial (una clave foránea debe corresponder a una clave primaria existente) y la integridad de dominio (los valores de los atributos deben respetar sus dominios).

#### Operaciones del Modelo Relacional

1. **Selección (Select)**
   * Extrae filas de una tabla que cumplen con ciertos criterios.
   * **Ejemplo**:

SELECT \* FROM Empleados WHERE Edad > 30;

1. **Proyección (Project)**
   * Extrae ciertas columnas de una tabla, eliminando las demás.
   * **Ejemplo**:

SELECT Nombre, Edad FROM Empleados;

1. **Unión (Join)**
   * Combina filas de dos o más tablas basadas en una condición común.
   * **Ejemplo**:

SELECT Empleados.Nombre, Departamentos.Nombre

FROM Empleados

JOIN Departamentos ON Empleados.DeptID = Departamentos.ID;

1. **Unión (Union)**
   * Combina las filas de dos tablas para crear una nueva tabla, eliminando duplicados.
   * **Ejemplo**:

SELECT Nombre FROM Empleados

UNION

SELECT Nombre FROM Clientes;

1. **Diferencia (Difference)**
   * Devuelve las filas presentes en una tabla pero no en otra.
   * **Ejemplo**:

SELECT Nombre FROM Empleados

EXCEPT

SELECT Nombre FROM Clientes;

1. **Producto Cartesiano (Cartesian Product)**
   * Combina todas las filas de dos tablas.
   * **Ejemplo**:

SELECT \* FROM Empleados, Proyectos;

#### Ventajas del Modelo Relacional

1. **Simplicidad**
   * El modelo relacional es intuitivo y fácil de entender. La organización de datos en tablas con filas y columnas es natural y facilita su manipulación.
2. **Flexibilidad**
   * Las relaciones entre tablas pueden ser modificadas y extendidas sin afectar a las aplicaciones existentes, permitiendo una evolución sencilla de la base de datos.
3. **Integridad y Seguridad**
   * Las reglas de integridad garantizan la consistencia de los datos. Los SGBD relacionales proporcionan mecanismos robustos de seguridad para controlar el acceso a los datos.
4. **Acceso Eficiente a los Datos**
   * Los SGBD relacionales están optimizados para ejecutar consultas complejas de manera eficiente, utilizando índices y otras técnicas de optimización.
5. **Estándares de Lenguaje de Consulta**
   * SQL (Structured Query Language) es el lenguaje estándar para interactuar con bases de datos relacionales, proporcionando un método consistente para definir, manipular y controlar los datos.

#### Desafíos del Modelo Relacional

1. **Escalabilidad**
   * Las bases de datos relacionales pueden enfrentar dificultades para escalar horizontalmente, aunque las técnicas de particionamiento y replicación pueden mitigar este problema.
2. **Complejidad en el Diseño**
   * El diseño adecuado de una base de datos relacional puede ser complejo y requiere una planificación cuidadosa para asegurar la normalización y evitar la redundancia de datos.
3. **Coste de Implementación**
   * Los sistemas de gestión de bases de datos relacionales pueden ser costosos en términos de licencias, hardware y mantenimiento.

#### Conclusión

El modelo relacional ha demostrado ser una herramienta poderosa y flexible para el modelado y la gestión de bases de datos. Su capacidad para organizar datos de manera estructurada y consistente, junto con su soporte para operaciones complejas y consultas eficientes, lo convierte en la base de la mayoría de los sistemas de gestión de bases de datos utilizados hoy en día. A pesar de los desafíos asociados, las ventajas del modelo relacional en términos de integridad, flexibilidad y eficiencia lo mantienen como una elección preferida para muchas aplicaciones empresariales y de desarrollo de software.

|  |  |
| --- | --- |
| año | Evento Clave |
| 1970 | Edgar F. Codd publica el artículo "A Relational Model of Data for Large Shared Data Banks", introduciendo el modelo relacional. |
| 1974 | IBM desarrolla el lenguaje de consulta SEQUEL, precursor de SQL. |
| 1979 | Oracle lanza el primer SGBD relacional comercial basado en SQL. |
| 1986 | ANSI e ISO estandarizan SQL como el lenguaje para bases de datos relacionales. |
| 1989 | Se publican las reglas de integridad referencial y la normalización de SQL por ANSI. |
| 1995 | MySQL se lanza como un sistema de gestión de bases de datos relacional de código abierto. |
| 2008 | Introducción de bases de datos NoSQL, ofreciendo alternativas para grandes volúmenes de datos no estructurados, pero el modelo relacional sigue siendo dominante para datos estructurados. |