

Bases de datos

Hernández Pérez Yair Edwin
Facultad de Ingeniería
Universidad Nacional Autónoma de México
Cdmx, México
yairedwinhp16@gmail.com

Index Terms—12 reglas de codd

A. INTRODUCCIÓN

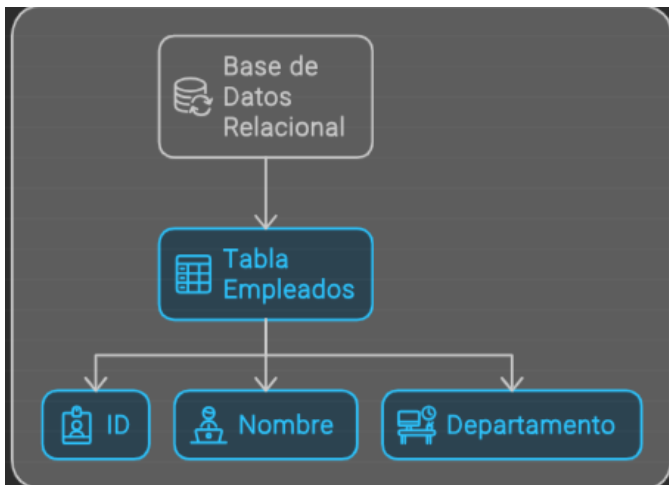
Esta es una investigación sobre las reglas de Codd las 12 reglas de Codd, propuestas por Edgar F. Codd, son un conjunto de principios diseñados para definir qué características debe tener un sistema de gestión de bases de datos para ser considerado verdaderamente relacional. Estas reglas buscan asegurar que las bases de datos relacionales mantengan su integridad y funcionalidad.

I. REGLA DE LA INFORMACIÓN

Toda la información en una base de datos relacional se representa explícitamente en tablas.

Ejemplo:

En una base de datos de empleados, toda la información sobre los empleados se almacena en una tabla llamada Empleados con columnas como ID, Nombre, Departamento, etc.



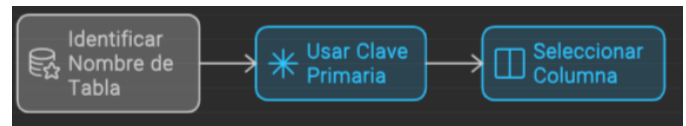
II. REGLA DEL ACCESO GARANTIZADO

Cada dato debe ser accesible mediante una combinación de nombre de tabla, clave primaria y nombre de columna.

Ejemplo:

Para acceder al nombre de un empleado con ID = 123, se

puede usar una consulta como `SELECT Nombre FROM Empleados WHERE ID = 123`.



III. TRATAMIENTO SISTEMÁTICO DE VALORES NULOS

Los valores nulos deben ser soportados para representar información desconocida o inaplicable.

Ejemplo:

Si un empleado no tiene un segundo nombre, el campo SegundoNombre puede contener un valor nulo.

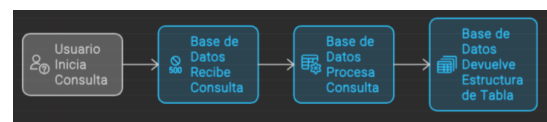


IV. CATÁLOGO DINÁMICO EN LÍNEA BASADO EN EL MODELO RELACIONAL

La estructura de la base de datos debe ser accesible y manipulable usando el mismo lenguaje que los datos.

Ejemplo:

Los usuarios pueden consultar la estructura de la tabla Empleados usando una consulta como `SELECT * FROM "INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME" = 'Empleados'`.

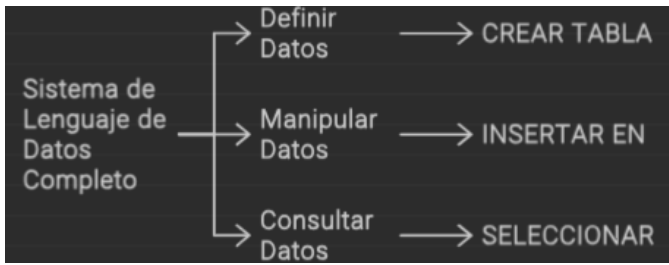


V. REGLA DEL SUBLenguaje de Datos Completo

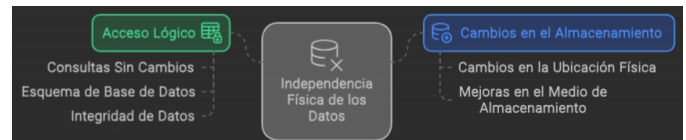
El sistema debe soportar un lenguaje de datos completo que permita definir, manipular y consultar datos.

Ejemplo:

SQL es un lenguaje que permite realizar estas operaciones, como `CREATE TABLE`, `INSERT INTO`, `SELECT`, etc.



Si se cambia la ubicación física de la tabla Empleados, las consultas como `SELECT * FROM Empleados` deben seguir funcionando sin cambios.



VI. REGLA DE ACTUALIZACIÓN DE VISTAS

Las vistas deben ser actualizables.

Ejemplo:

Si se crea una vista `VistaEmpleados` que muestra solo los empleados del departamento de ventas, debería ser posible actualizar los datos a través de esta vista.



IX. INDEPENDENCIA LÓGICA DE LOS DATOS

Los cambios en el esquema lógico no deben afectar las aplicaciones que usan los datos.

Ejemplo:

Si se añade una nueva columna `FechaContratacion` a la tabla `Empleados`, las consultas existentes que no usan esta columna no deben verse afectadas.



X. INDEPENDENCIA DE LA INTEGRIDAD

Las restricciones de integridad deben ser definibles en el sublenguaje de datos y almacenadas en el catálogo.

Ejemplo:

Definir una restricción de clave primaria en la tabla `Empleados` usando `ALTER TABLE Empleados ADD CONSTRAINT PK_Empleados PRIMARY KEY (ID)`.



VII. INSERCIÓN, ACTUALIZACIÓN Y ELIMINACIÓN DE ALTO NIVEL

El sistema debe soportar operaciones de inserción, actualización y eliminación que afecten múltiples filas y/o tablas a la vez.

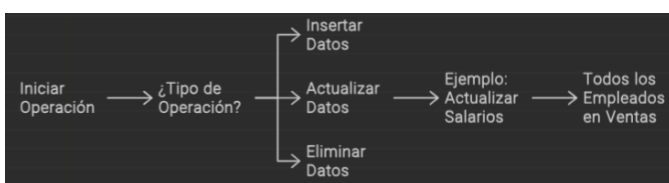
Ejemplo:

Una consulta como `UPDATE Empleados SET Salario = Salario * 1.1 WHERE Departamento = 'Ventas'` actualiza los salarios de todos los empleados del departamento de ventas.

VIII. INDEPENDENCIA FÍSICA DE LOS DATOS

Los cambios en el almacenamiento físico de los datos no deben afectar el acceso lógico a los mismos.

Ejemplo:

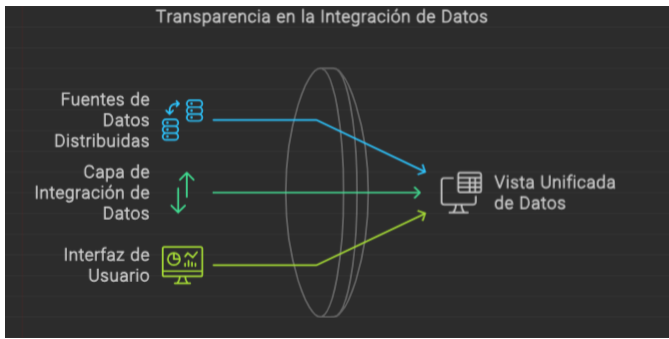


XI. INDEPENDENCIA DE LA DISTRIBUCIÓN

La distribución de los datos en múltiples ubicaciones debe ser transparente para el usuario.

Ejemplo:

Consultar datos de empleados distribuidos en diferentes servidores como si estuvieran en una sola tabla.



XII. REGLA DE NO SUBVERSIÓN

Si el sistema tiene un lenguaje de bajo nivel, este no debe poder subvertir las reglas de integridad y seguridad.

Ejemplo:

No permitir que un lenguaje de bajo nivel omita las restricciones de integridad definidas en SQL.



REFERENCES

- [1] Codd, E. F. (1990). The Relational Model for Database Management: Version 2. Addison-Wesley.
- [2] Date, C. J. (2003). An Introduction to Database Systems (8th ed.). Addison-Wesley.