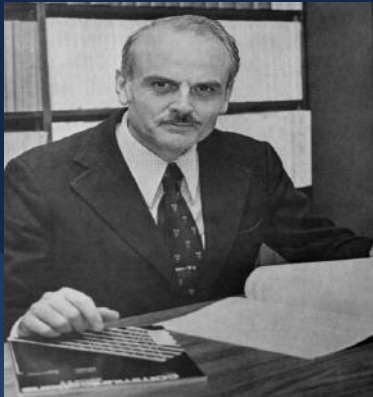


# Las 12 reglas de Codd



# Modelo Relacional

Es un modelo lógico que establece una estructura sobre los datos y permite representarlos en forma de tablas. Fue introducido en 1970 por Edgar Frank Codd (empleado de IBM) en su documento A Relational Model of Data for Large Shared Data Banks y se basa en 2 principales conceptos matemáticos



For more info:

<https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>



01

Lógica de predicados: para conocer la veracidad a través del uso de aserciones.



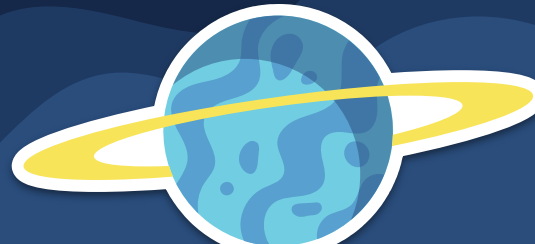
02

Teoría de conjuntos: para realizar la manipulación de los datos.



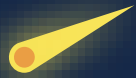
# Las 12 reglas de Codd

Sirven como una guía para evaluar si un sistema de gestión de bases de datos (DBMS) cumple con los principios fundamentales de una base de datos relacional.










Regla 0: Regla fundamental.

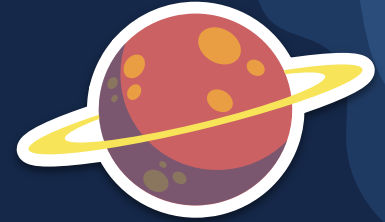
Todo sistema que se defina como 

◊ sistema de gestión de base de  
 datos relacional, ha de poder  
gestionar las bases de datos  
exclusivamente con sus capacidades  
relacionales. ✨

Regla 1: 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. 



Si tienes una tabla llamada "Clientes" con las columnas "ID\_Cliente", "Nombre", y "Correo electrónico", podrías tener datos como:



sql

ID_Cliente	Nombre	Correo electrónico
1	Juan Pérez	juan@email.com
2	María López	maria@email.com
3	Carlos Ruiz	carlos@email.com

## Regla 2: Regla del acceso garantizado.

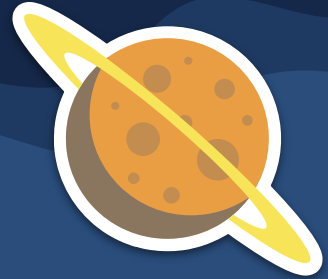
Se garantiza que todos y cada uno de los datos (valor atómico) de una base de datos relacional  son accesibles lógicamente mediante una  combinación de nombre de tabla, valor de clave primaria y nombre de columna.



Si deseas acceder a la información del cliente con ID\_Cliente igual a 2, podrías usar la consulta SQL:

sql

```
SELECT * FROM Clientes WHERE ID_Cliente = 2;
```



**Regla 3:** 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) .



Supongamos que algunos clientes no han proporcionado su dirección. Puedes representar esto con un valor nulo en la columna "Dirección":



```
sql Copy
```

ID_Cliente	Nombre	Correo electrónico	Dirección
1	Juan Pérez	juan@email.com	Calle A, Ciudad
2	María López	maria@email.com	NULL
3	Carlos Ruiz	carlos@email.com	Calle B, Ciudad

## Regla 4: Catálogo dinámico en línea basado en el modelo relacional.

El sistema debe soportar un catálogo en línea, el catálogo relacional, que da acceso a la estructura de la base de datos y que debe ser accesible a los usuarios autorizados.





Supongamos que queremos obtener información sobre las columnas de la tabla "Clientes".

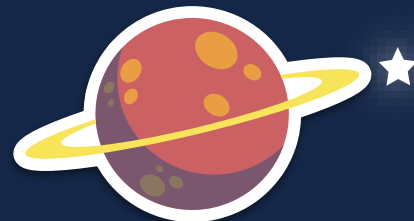


Podríamos consultar el diccionario de datos de la base de datos para obtener esta información:



sql

```
DESCRIBE Clientes;
```



Esta consulta hipotética podría devolver

- información sobre las columnas, como sus nombres, tipos de datos y restricciones.

Regla 5: Regla del sublenguaje de  
datos completo. El sistema debe  
proporcionar al menos un lenguaje de datos  
completo que pueda ser utilizado para  
definir datos, relaciones, restricciones de  
integridad y manipulaciones de datos.



\* Imaginemos que queremos realizar una operación más compleja en nuestra base de datos, como encontrar todos los clientes que han realizado una compra en los últimos 30 días. Podríamos usar SQL para expresar esta operación:



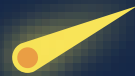
sql

```
SELECT * FROM Clientes c
JOIN Compras co ON c.ID_Cliente = co.ID_Cliente
WHERE co.FechaCompra >= CURDATE() - INTERVAL 30 DAY;
```

Esta consulta utiliza un lenguaje de datos completo (SQL) para expresar la operación que deseamos realizar.

# Regla 6: Regla de actualización de vistas.

- ♦ Todas las vistas que son teóricamente actualizables son también actualizables por el sistema.





\* Supongamos que tenemos una vista llamada "ClientesVIP" que muestra solo a los clientes que han realizado compras por un monto superior a cierta cantidad. Podríamos actualizar la información de un cliente VIP directamente a través de esta vista:



sql

```
UPDATE ClientesVIP SET Saldo = 1500 WHERE ID_Cliente = 1;
```

- Esta actualización a través de la vista garantiza que se respeten las restricciones definidas en la vista.

# Regla 7: Inserción, actualización y borrado de alto nivel



El sistema debe permitir la manipulación de alto nivel en los datos, es decir, sobre conjuntos de tuplas.



Supongamos que queremos obtener información sobre todas las tablas presentes en nuestra base de datos. Podríamos consultar el catálogo de la base de datos para obtener esta información:

sql

```
SELECT * FROM Information_Schema.Tables WHERE  
Table_Schema = 'nombre_de_tu_base_de_datos';
```

Esta consulta, aunque es hipotética y puede variar según la implementación del sistema de gestión de base de datos, mostraría información sobre las tablas presentes en la base de datos.

# Regla 8: 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. ✦

Imaginemos que tenemos una vista llamada "ClientesVIP" que muestra solo a los clientes cuyo saldo es mayor a cierta cantidad. Si intentamos insertar un nuevo cliente a través de esta vista y el saldo del nuevo cliente no cumple con la condición de la vista, se produciría un error para mantener la integridad de la vista:

sql

Copy code

```
INSERT INTO ClientesVIP (ID_Cliente, Nombre, Saldo) VALUES (4, 'Ana García', 100);  
-- Esto generaría un error si el saldo no cumple con la condición de la vista.
```

La regla de integridad de vista garantiza que las inserciones y actualizaciones respeten las condiciones establecidas en la vista.

## Regla 9: 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. ✦

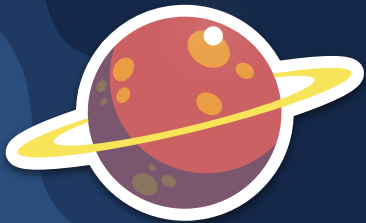
Supongamos que queremos obtener información sobre un cliente específico sin conocer todas las tablas y relaciones internas. Podríamos hacer esto usando una consulta simple:

```
sql
```

```
SELECT * FROM Clientes WHERE Nombre = 'Carlos Ruiz';
```

Esta consulta nos permite buscar información sobre un cliente sin necesidad de conocer la estructura interna completa de la base de datos.

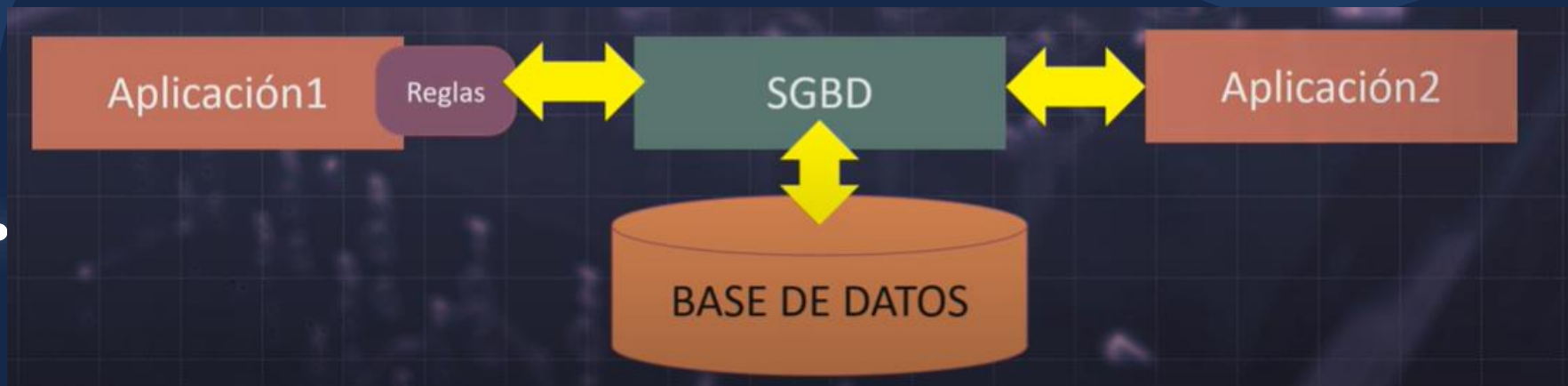
\* La independencia de datos lógica  
es más difícil de lograr que la  
independencia física de datos.





# Regla 10: Independencia de la integridad.

- ✧ Las restricciones de integridad se deben especificar por separado de los programas de aplicación y almacenarse en la base de datos. Debe ser posible cambiar esas restricciones sin afectar innecesariamente a las aplicaciones existentes.



# Regla 11: Independencia de la

\*distribución. El usuario final no ha de ver que

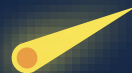
- \* los datos están distribuidos en varias ubicaciones. Los usuarios deben tener siempre la impresión de que los datos se encuentran en un solo lugar. ♦



## Regla 12: La regla de la no

- \* subversión. Si el sistema proporciona una
- \* interfaz de bajo nivel de registro, además de ♦ una interfaz relacional, que esa interfaz de bajo nivel no se pueda utilizar para subvertir el sistema (sin pasar por alto la seguridad relacional o limitación de integridad, por ejemplo).





## Preguntas (Opcional )





## Bibliografía

Alex. (s/f). MySQL "DESCRIBE TABLE" en PostgreSQL. Netveloper.com. Recuperado el 29 de febrero de 2024, de <https://www.netveloper.com/mysql-describe-table-en-postgres>

Astudillo, D. [@danielastudillo3259]. (2016, septiembre 11). Explicación de las 12 reglas de codd. Youtube. <https://www.youtube.com/watch?v=0l0165WhTyM>

Bpc, G. V.-C. S. (2013, julio 18). 12 reglas de Codd para bases de datos Relacionadas. Medievals Trucos. <https://medievalstrucos.com/2013/07/18/12-reglas-de-codd-para-bases-de-datos-relacionadas/>

DBA. (2017, marzo 2). Reglas de Codd de las bases de datos relacionales. DBA dixit; DBA. <https://dbadixit.com/reglas-codd-las-bases-datos-relacionales/>

Sanchez, J. [@JorgeSancheznet]. (2018, junio 11). Las 12 Reglas de Codd. Youtube. <https://www.youtube.com/watch?v=HkgVLS8braQ>