//Clase N°1

Condiciones de la integridad:

1. **Restricción de integridad de usuario**, es decir, condiciones específicas de una BD concreto, pero no son necesariamente relevantes en otras.

2. **Restricción de integridad de modelo**, ósea Condiciones propias de un modelo de datos. (se cumplen en todas las bases de datos del modelo).

Reglas de integridad:

Regla de integridad de unicidad de clave primaria => las PK deben ser únicas e irrepetibles. Entonces la extensión de R no puede tener en ningún momento dos tuplas con la misma combinación de valores para los atributos de CP.

Regla de integridad de entidad de clave primaria =>

Los atributos PK de una entidad != null (necesario para distinguir entre tuplas y R).

Regla de integridad referencial => FK ósea entidad1(FK) = entidad2(PK).

Restricción => En caso de borrado es si la tupla tiene una PK referenciada por una FK no permite ser borrada

Actualización en cascada => La actualización en cascada consiste en permitir la operación de actualización de la tupla, y en efectuar operaciones compensatorias que propaguen en cascada la actualización a las tuplas que la referenciaban.

Anulación => Setear valores a nulos.

Regla de integridad de dominio => (noción de dominio) 2 cuestiones: 1°.
> Un valor no nulo de atributo Ai debe pertenecer al dominio Ai.
> Operadores que pueden aplicarse sobre los valores dependen de los dominios de estos valores.

Data Types => Numbers (Integer); **Alphanumeric (**String); **Date**;

Atributos de campo => Null || Not null || Default.

Prepared Statements => Sintaxis en la que se deja valores sin especificar usando un "?" para llenarlo después.

ResultSet => Un ResultSet, o conjunto de resultados, contiene los resultados de una consulta SQL, por lo que es un conjunto de filas obtenida desde una base de datos, así como Metadatos sobre la consulta y los tamaños de cada columna.

ORM (Object Relational Mapping) => Permite mapear los datos de los objetos en un formato correcto para guardar la información en una DB creándose una VDB donde los datos de nuestra aplicación quedan vinculados a la DB (Persistencia).

//CLASE 2

DB Distribuida => Base de datos construida sobre una red de computadoras, la info está distribuida a través de varios sitos en la red y se acceden de puntos geográficos diferentes.
Por ende, una Base de Datos Distribuida es una colección de datos que pertenecen lógicamente a un solo sistema, pero se encuentra físicamente distribuido.

Bases de esta wea:

1. Autonomía local =>
   Operaciones de un lugar se gestionan en el mismo (autonomía).
2. No dependencia de un sitio central =>
   No está copado porque trae Bottle Neck || Vulnerabilidad.
3. Operación continúa =>
   El sistema debería ser continuo = 24/7 = no apagarse “Debería”.
4. Independencia con respecto a la localización =>
   Mas de lo mismo.
5. Independencia con respecto a la fragmentación =>
   Desconocer la fragmentación que usa la DB.
6. Independencia de réplica =>
   Se pueden hacer bakcups.
7. Procesamiento distribuido de consultas =>
   Separar las instrucciones para mejor manejo.
8. Manejo distribuido de transacciones =>
   Control de recuperación y el control de concurrencia.
9. Independencia con respecto al equipo =>
   El SGBD se debe poder ejecutar where ever.
10. Independencia con respecto al sistema operativo =>
    Same que arriba but SO.
11. Independencia con respecto a la red =>
    Tiene que andar != topología y tecnología de la red.
12. Independencia con respecto al SGBD =>
    Los SGBD en los diferentes sitios pueden manejar otra interfaz.

//Clase N°6

Bases de Datos NoSQL >

RESUMEN:
Las bases de datos NoSQL han experimentado un importante incremento en su aplicación en los últimos tiempos. La gran flexibilidad que ofrecen y las posibilidades que brindan desde el punto de vista de la optimización en sus diseños de acuerdo al problema a resolver las convierten en una atractiva variante a tener en cuenta para los desarrolladores de aplicaciones de gestión de información. En el presente artículo se hace un recorrido por la evolución de los tipos de bases de datos hasta llegar a las relacionales, las cuales se analizan con el objetivo de mostrar los aspectos asociados a estas que propiciaron el surgimiento de las NoSQL.

Key words: relational databases, NoSQL databases, digital management information.

DB no Relacional => Base de datos en dónde los datos no se organizan en tablas, sino que utilizan otros métodos como jerarquía, usar columnas en vez de filas, agrupar en grafos, entre otras. **DB Orientada a documentos =>** Una base de datos orientada a documentos o almacén de documentos es un tipo de BD NoSQL que sirve para gestionar información como lo son documentos o datos semi estructurados. A diferencia de las bases de datos relacionales y sus "tablas", los sistemas documentales están diseñados entorno a la definición abstracta de un "documento".
JsON=> JavaScript Object Notation, es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías.
DB orientada a grafos => Las bases de datos orientadas a grafos representan la información como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar teoría de grafos para recorrer la base de datos ya que esta puede describir atributos de los nodos (entidades) y las aristas (relaciones).
Teorema de Brewer => El teorema de Brewer, o sistema CAP por sus siglas en inglés: Consistency, Availability y Partition Tolerance.
Hace referencia a que en los sistemas distribuidos solo se puede garantizar dos de las tres cuestiones mencionadas anteriormente, por lo tanto, se debe tener en cuenta que es lo más importante para nuestra base de datos para optar por un sistema acorde a nuestras necesidades.

//Clase N°8
Estructura Centralizada VS Sistema Cliente-Servidor.
En la estructura centralizada una misma CPU realiza las funciones de las distintas terminales que pueden estar conectadas a esta. A diferencia de esto el sistema Cliente-Servidor hace uso de un servidor que brinda servicios a diferentes dispositivos.

/*Las muchas cosas que iban en el diome y me dió paja resumir, quizás mañana...*/

En arquitectura compartida =>

Memoria compartida: Todos los procesadores comparten una memoria común.
Disco compartido: Todos los procesadores comparten un conjunto de discos en común (o agrupaciones). Algunas veces los sistemas de disco compartido se denominan agrupaciones.
Sin compartimiento: Los procesadores no comparten ni memoria ni disco.
Jerárquica: Este modelo es un híbrido de las arquitecturas anteriores.