IIC2413

AYUDANTÍA 13

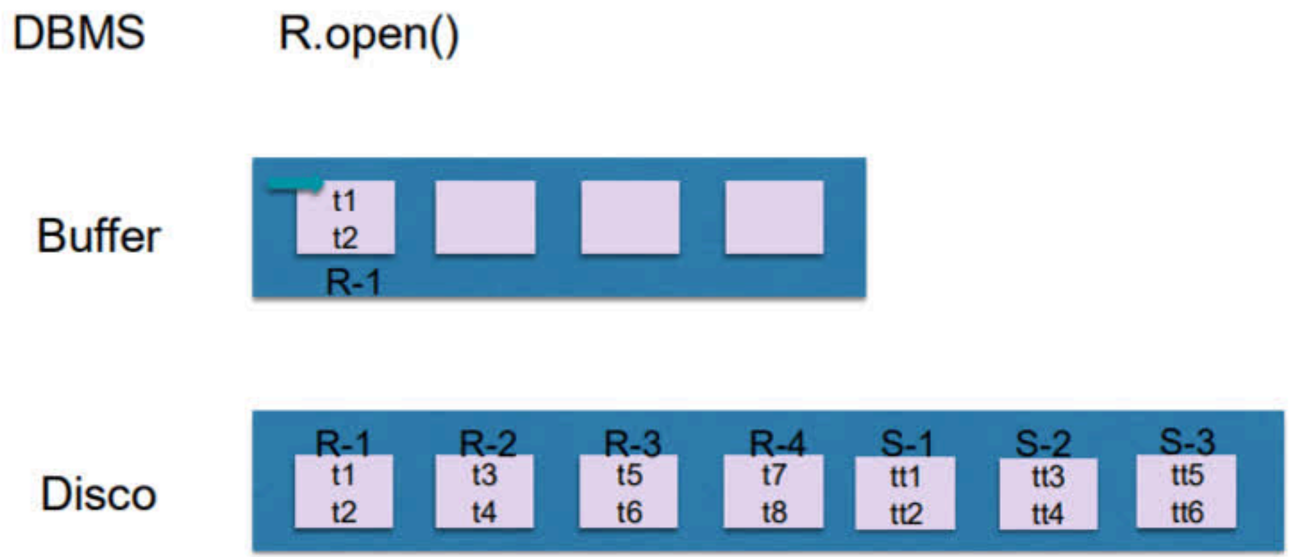
REPASO EXAMEN

Iterador lineal

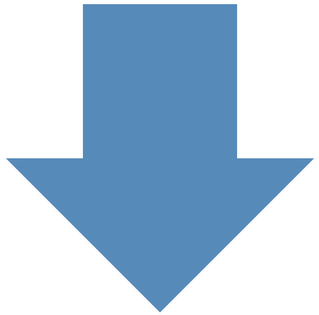
Para una relación R:

R.open() – se posiciona **antes** de la primera tupla de R
R.next() – devuelve la siguiente tupla o NULL
R.close() – cierra el iterador

Páginas, disco y buffer



El flujo principal de un proceso lineal en la gran mayoría de los casos sigue el siguiente formato



```
R.open()

 t:= R.next()
while t != null do
            output t
    t:= R.next()


R.close()
```

Operador lineal v/s Iterador lineal (Ej: Selección sin indice)

Lógica del operador lineal (define sus propios métodos)

```
open()          ← open() del operador
    R.open()    ← open() del iterador

next()          ← next() del operador
    t:= R.next()    ← next() del iterador
    while t != null do
        if t satisface condición then
            return t
        t:= R.next()    ← next() del iterador
    return null
close()         ← close() del operador
    R.close()   ← close() del iterador
```

Llamado al operador lineal

← = función del operador

Para recorrer la selección Sel = σ_{cond} (**R**)

```
Sel.open()  ←

t := Sel.next()  ←
while t != null do
        output t
        t:= Sel.next()  ←
Sel.close()  ←
```

Pregunta 1: Examen 2025-1

Pregunta 2 - Evaluación de consultas (15 pts)

Se tienen las siguientes 2 relaciones:

- Pedidos(id_pedido, fecha_pedido, monto, estado)
- Despachos(id_despacho, id_pedido, fecha_despacho)

Cada página puede almacenar 20 tuplas de **Pedidos** y 50 tuplas de **Despacho**. Además el buffer almacena 12 páginas.

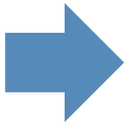
a) (4 pts.) ¿Cuántas lecturas de páginas se requieren para realizar un Nested Loop Join entre **Pedidos** y **Despachos** usando esta tabla como relación externa?

***Hint: Flujo de Nested Loop Join:**

```
open()
    R.open()
    S.open()
    r:= R.next()

close()
    R.close()
    S.close()
```

```
next()
    while r != null do
        s:= S.next()
        if s == null then
            S.close()
            r:= R.next()
            S.open()
        else if (r, s) satisfacen p then
            return (r, s)
    return null
```



$Join = R \bowtie_p S$

```
Join.open()

 t:= Join.next()
while t != null do
            output t
   t:= Join.next()

Join.close()
```

Solución

Solución:

$$\text{Costo NLJ} = \text{Pág. de Pedidos} + (\text{Tuplas de Pedidos} \times \text{Pág. de Despachos})$$

$$\text{Pág. de Pedidos} = \frac{\text{Tuplas de Pedidos}}{20}$$

$$\text{Pág. de Despachos} = \frac{\text{Tuplas de Despachos}}{50}$$

Para cada tupla de **R** debemos leer **S** entera, aparte de leer **R** entera una vez

Costo en I/O es:

$$\text{Costo}(\mathbf{R}) + \text{Tuplas}(\mathbf{R}) \cdot \text{Costo}(\mathbf{S})$$

Pregunta 2

b) Usando la tabla anterior y con buffer de tamaño 12 ¿Cuál sería el costo de lectura para realizar un Block Nested Loop Join entre Pedidos y Despachos?

***Hint: Flujo de Block Nested Loop Join:**

```
open()
    R.open()
    fillBuffer()

close()
    R.close()
    S.close()
```

```
next()
    while Buff != empty do
        while s != null do
            r:= Buffer.next()
            if r == null then
                Buffer.reset()
                s:= S.next()
            else if (r,s) satisfacen p then
                return (r,s)
        fillBuffer()
    return null
```

```
fillBuffer()
    Buff = empty
    r:= R.next()
    while r != null do
        Buff = Buff union r
        if Buff.isFull() then
            break
        r:= R.next()
    S.open()
    S.next()
```



$Join = R \bowtie_p S$

```
Join.open()

 t:= Join.next()
while t != null do
            output t
    t:= Join.next()

Join.close()
```

Solución

Por cada vez que llenamos el buffer recorremos a **S** entera una vez

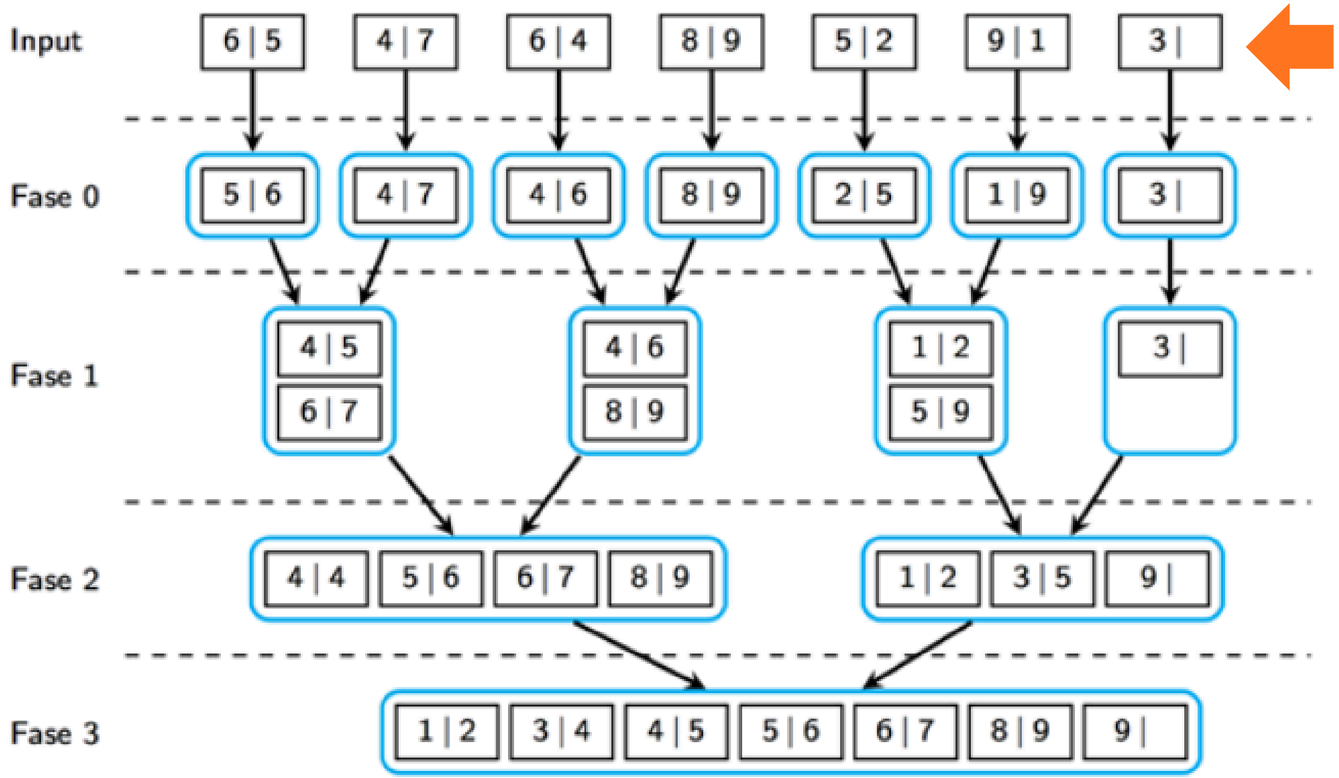
Costo en I/O es:

$$\text{Costo}(\mathbf{R}) + (\text{Páginas}(\mathbf{R})/(\text{Buffer-2}))\cdot\text{Costo}(\mathbf{S})$$

$$\text{Costo BNLJ} = Pág.\ de\ Pedidos\ +\ \frac{Pág.\ de\ Pedidos}{Buffer-2} \times\ Pág.\ de\ Despachos$$

Buffer = 12

External Merge Sort



Esto es un “run” de tamaño 1

- Por cada fase tomamos varios runs ya ordenados y los fusionamos siguiendo la lógica de MergeSort, repitiendo el proceso hasta que todos los runs se combinan en un único archivo completamente ordenado.

Tipos de datos

Dependiendo del formato en que guardan la información

- **Estructurados**: SQL, Excel
- **Semiestructurados**: JSON, Grafos
- **No Estructurado**: Texto Plano

Dependiendo de como se relacionan los datos entre sí y como se organizan

SQL

- Esquema definido
- ACID
- Modelos Relacionales

No SQL

- Sin esquema
- Menos consistentes
- Tipos: Key-value, grafos, documentos

Key-Value

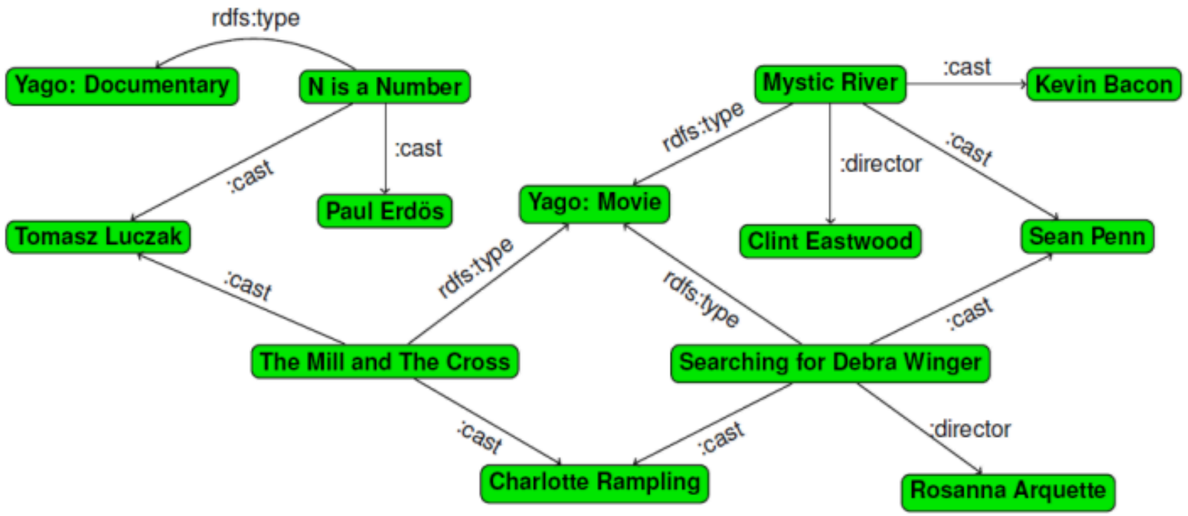
- Pares simples
- Búsqueda por clave
- No soporta relaciones internas
- Extremadamente rápido (casi instantánea)
- Ej: Redis, DynamoDB, etc...

Orientada a documentos

- El valor es un documento
- Búsqueda por campos, índices, filtros
- Puede contener subdocumentos o listas
- Rápido pero más lento que Key-Value
- Ej: MongoDB, CouchDB, etc...

Grafos

- Los nodos son entidades/atributos
- Las aristas son las relaciones entre los nodos



MongoDB

- Base de datos orientada a documentos JSON y no SQL

- Colección = Conjunto de documentos pertenecientes a la misma entidad

Código

```
show dbs  … muestra bases de datos disponibles
use dbName  … ahora usamos base de datos dbName
show collections  … colecciones en nuestra base de datos

db.colName.find()  … todos los documentos en la colección colName

db.colName.find().pretty()  … muentra los datos de mejor forma

db.colName.find({"name": "Adrian"})  … selección

db.colName.find({"age": {$gte:23}})  … selección

db.colName.find({"age": {$gte:23}},{"name":1})  … proyección
```

Pregunta 3: Examen 2025-1

b) (2 pts.) ¿Cuál de la siguiente opción es falsa con respecto a la BD Documentales?

I. Usan principalmente documentos JSON

II. Ideales para realizar cruce de datos complejos

III. Lenguaje de consulta poderoso

IV. No necesitan respetar un esquema de datos

Solución

Solución: La opción que es falsa es la II. Las BD documentales no están ideadas para realizar cruces de datos (joins) complejos. Esto debido a que esta lógica no la soportan y se debe realizar en el código de la aplicación.

Pregunta 4

Que operación realizan exactamente las siguientes lineas de código en MongoDB

- ```
  db.users.find({ "status": "active" })
  ```
- ```
  db.orders.find({ "total": { $gt: 10000 } })
  ```
- ```
  db.people.find({ "age": { $lte: 30 } },
    { "name": 1, "age": 1, "_id": 0})
  ```
  ```

Solución

- `db.users.find({ "status": "active" })`

Respuesta: Busca a todos los *usuarios* cuyo *status* sea **“active”** y los retorna

- `db.orders.find({ "total": { $gt: 10000 } })`

Respuesta: Encuentra todas las *ordenes* cuyo *total* sea **“gt” (greater than)** 10000 y los retorna

- ```
  db.people.find({ "age": { $lte: 30 } },
  { "name": 1, "age": 1, "_id": 0})
  ```

Respuesta: Busca a todas las *personas* cuya *edad* sea **“lte”(lower than or equal)** a 30. Solo retorna los atributos **“name”** y **“age”** sin incluir **“_id”**
  ```

Full-text search & retrieval

Permite realizar búsquedas “inteligentes” de palabras en texto libre

TF
(Team Frequency)

FD(t) = Número de veces que aparece t en D dividido por el total de palabras en D

IDF
(Inverse Document Frequency)

IDF(t) = log(número de documentos / número de documentos en los que aparece t)

TF-IDF = FD(t) · IDF(t)

Pregunta 5: Examen 2024-2

Tienes un conjunto de documentos y deseas calcular el TF (Frecuencia de Término) y el IDF (Frecuencia Inversa de Documento) para cada término en los documentos.

- **Documento 1:** "El gato come pescado"
- **Documento 2:** "El perro come carne"
- **Documento 3:** "El gato y el perro son amigos"

a) Calcular el TF (Frecuencia de Término): para los términos gato y perro (0,1 pts)

b) Calcular el IDF (Frecuencia Inversa de Documento): para los términos gato y perro. Nota: $log(3/2) = 0{,}1761; log(3) = 0{,}4771$ (0.1 pts)

c) Calcular el TF-IDF: para los términos gato y perro. (0.1 pts)

Solución

a)

Solución: TF es Número de veces que aparece un Término en un documento, dividido por el número total de términos en el documento

Termino	gato	perro
Documento 1	1/4	0/4
Documento 2	0/4	1/4
Documento 3	1/7	1/7

b)

El IDF se calcula como el logaritmo del número total de documentos dividido por el número de documentos que contienen el término.

Termino	gato	perro
IDF	log(3/2)	log(3/2)

Solución

c)

El TF-IDF se calcula multiplicando el TF por el IDF.

Termino	gato	perro
Documento 1	0.25*0.1761=0.0440	0*0.1761=0
Documento 2	0*0.1761=0	0.25*0.1761=0.0440
Documento 3	0.14*0.1761=0.0246	0.14*0.1761=0.0246

Privacidad

Muchas bases de datos contienen información sensible o confidencial de personas.

Es nuestro deber hacer que estos datos sigan siendo privados.

¿Cómo preservamos la privacidad?

Técnicas de anonimización

Basadas en la generación de datos sintéticos

Permiten generar consultas sin pérdidas de datos importantes.

Técnicas de privacidad diferencial

Es un mecanismo que garantiza que cuando un algoritmo aleatorio opera sobre un conjunto de datos, no revele información específica de ningún individuo

Asegura que la probabilidad de obtener un resultado particular es casi la misma, independientemente de si un individuo está o no en el conjunto de datos.

Definiciones y convenciones

Identificador explícito (IdE)	Conjunto de atributos que individualmente identifican de manera explícita al "dueño" de un registro. Por ejemplo, nombre y RUT
Cuasi identificador (CId)	Conjunto de atributos que colectivamente tienen el potencial de identificar al dueño de un registro. Por ejemplo, día de nacimiento y dirección.
Atributos sensibles (AS)	Conjunto de atributos cuyo valor se desea proteger. Por ejemplo, religión, etnia, enfermedad, etc.
Atributos no-sensibles (AnS)	Conjunto de atributos que no tienen un carácter privado (y por lo tanto no se busca proteger). Por ejemplo, equipo de futbol o película favorita.

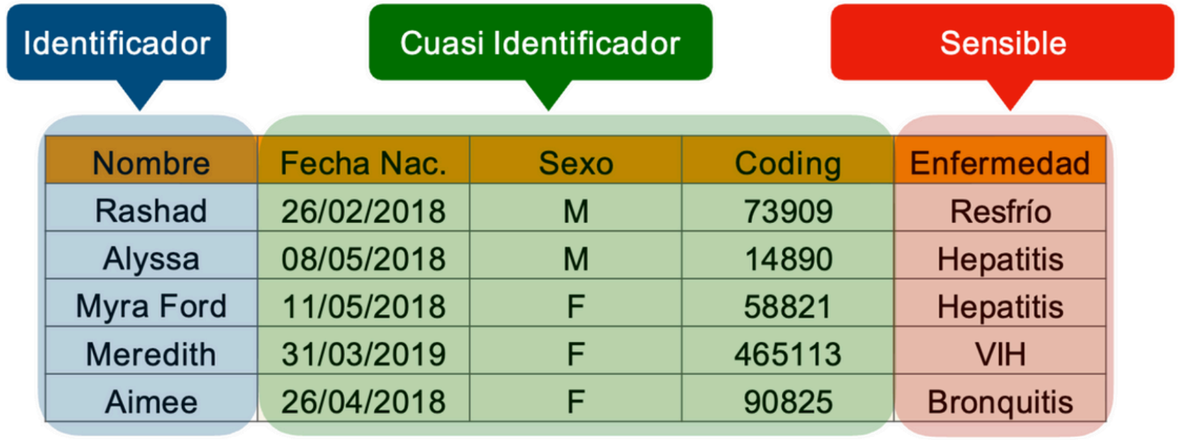


tabla resumen de las técnicas

Técnica	Descripción / Objetivo	Mecanismo Principal	Vulnerabilidades o Limitaciones
De-identificación / Anonimización	Proceso general para remover la asociación entre una persona y un conjunto de datos, buscando maximizar la utilidad y reducir riesgos.	Técnicas como supresión (borrar datos), variación (revolver datos) y enmascaramiento (ocultar caracteres) .	Es vulnerable a ataques de re-identificación, especialmente el ataque de asociación de registros usando datos externos.
k-Anonimato	Garantía formal que asegura que ningún individuo sea únicamente identificable, "escondiéndolo" en un grupo de al menos k personas.	Generalización de datos para que cada grupo de cuasi-identificadores (ej. edad, código postal) aparezca al menos k veces en la tabla.	Susceptible a ataques de homogeneidad (si todos en el grupo tienen el mismo dato sensible) y ataques usando datos auxiliares.
L-Diversidad	Extensión del k-anonimato que busca asegurar variabilidad en los datos sensibles dentro de cada grupo anónimo.	Requiere que para cada grupo de cuasi-identificadores, existan al menos L valores distintos para los atributos sensibles.	Aunque mejora el k-anonimato, sigue siendo susceptible a ataques si el adversario posee cierta información auxiliar específica o negativa sobre el grupo.
Privacidad Diferencial	Mecanismo matemático que garantiza que el resultado de un análisis sea prácticamente el mismo, esté o no incluido un individuo específico en la base de datos .	Agrega ruido aleatorio (ej. Mecanismo de Laplace) a las respuestas de las consultas. Se controla mediante un "presupuesto de privacidad" (ϵ).	Dificulta la interpretación directa debido al ruido y no suele usarse para generar datos sintéticos (tablas completas), sino para respuestas a consultas específicas.

Pregunta 3: Examen 2025-1

Pregunta 3 - Privacidad (15 pts)

a) (4 pts.) ¿Cuáles de las siguientes condiciones se deben cumplir para que un análisis de dato preserve la privacidad? Justifique su respuesta.

I. El análisis tiene k-anonimato mayor o igual a 50.

II. El análisis tiene l-diversidad.

III. No se viola la privacidad de algún individuo.

IV. Se puede aprender algo del análisis.

V. Usa Privacidad Diferencial.

Solución

Solución: Al analizar cada una de las afirmaciones tenemos lo siguiente:

- I: Aunque el k-anonimato es una medida que aumenta la privacidad, no lo asegura.
- II: Aunque l-diversidad es una medida que aumenta la privacidad, tampoco lo asegura.
- III: Esta es una condición que debe cumplirse, ya que asegura que no se ha violado la privacidad de ningún individuo.
- IV: Al cumplir la privacidad, debe ser factible el aprendizaje a partir de los datos. De nada sirve tener privacidad, sin aprender de los datos.
- V: La privacidad diferencial es un mecanismo que incrementa la privacidad, pero no es requisito, ya que se pueden usar otros mecanismos.

Por lo tanto la respuesta correcta es III y IV.

Pregunta 3: Examen 2025-1

(5 pts.) Relacione el concepto y su definición.

I. De-identificación (DeId)

II. Datos sintéticos (DS)

III. Re-identificación (ReId)

IV. k-Anonimato (kAn)

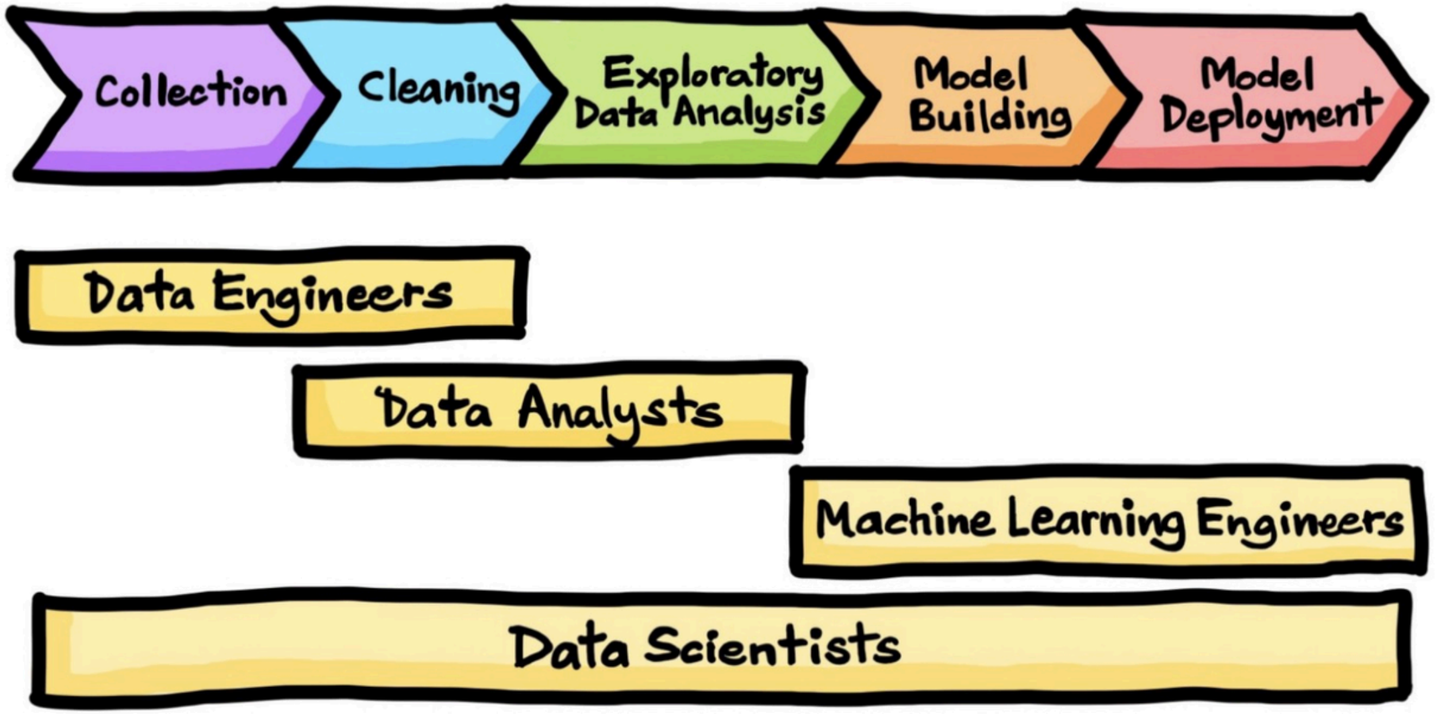
V. l-Diversidad (lDiv)

	Definición	**Concepto**
1	Para cada grupo de filas con los mismos CIds, aparece al menos un cierto número de veces repetidos.	
2	Para cada grupo de filas con los mismos CIds, por cada atributo sensible S, existen un cierto número de valores distintos.	
3	Es el proceso que vuelve a asociar una persona con un conjunto de datos.	
4	Es el proceso que remueve la asociación entre una persona y un conjunto de datos.	
5	Son datos generados artificialmente, como forma opuesta a recolectarlos del mundo real.	

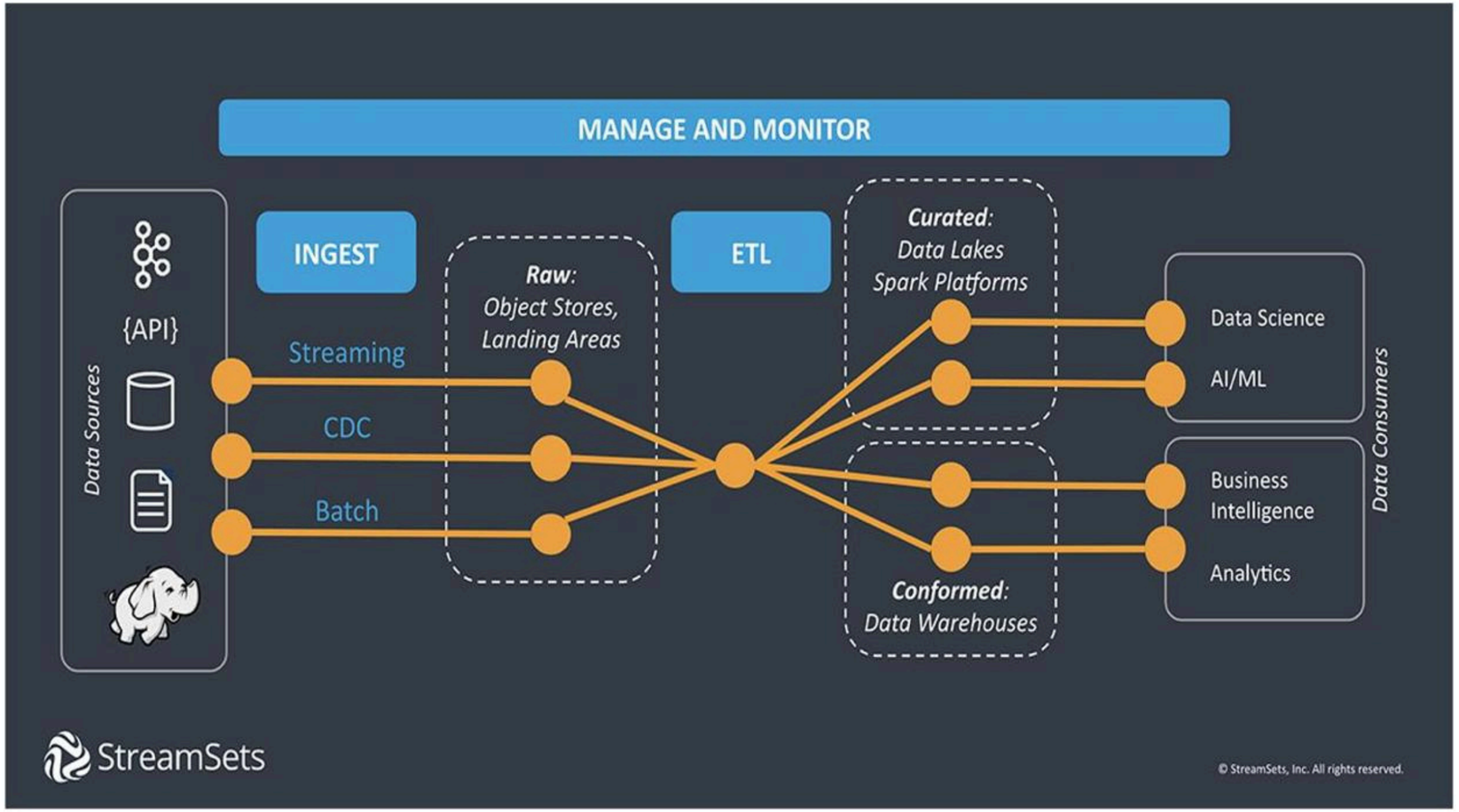
Solución

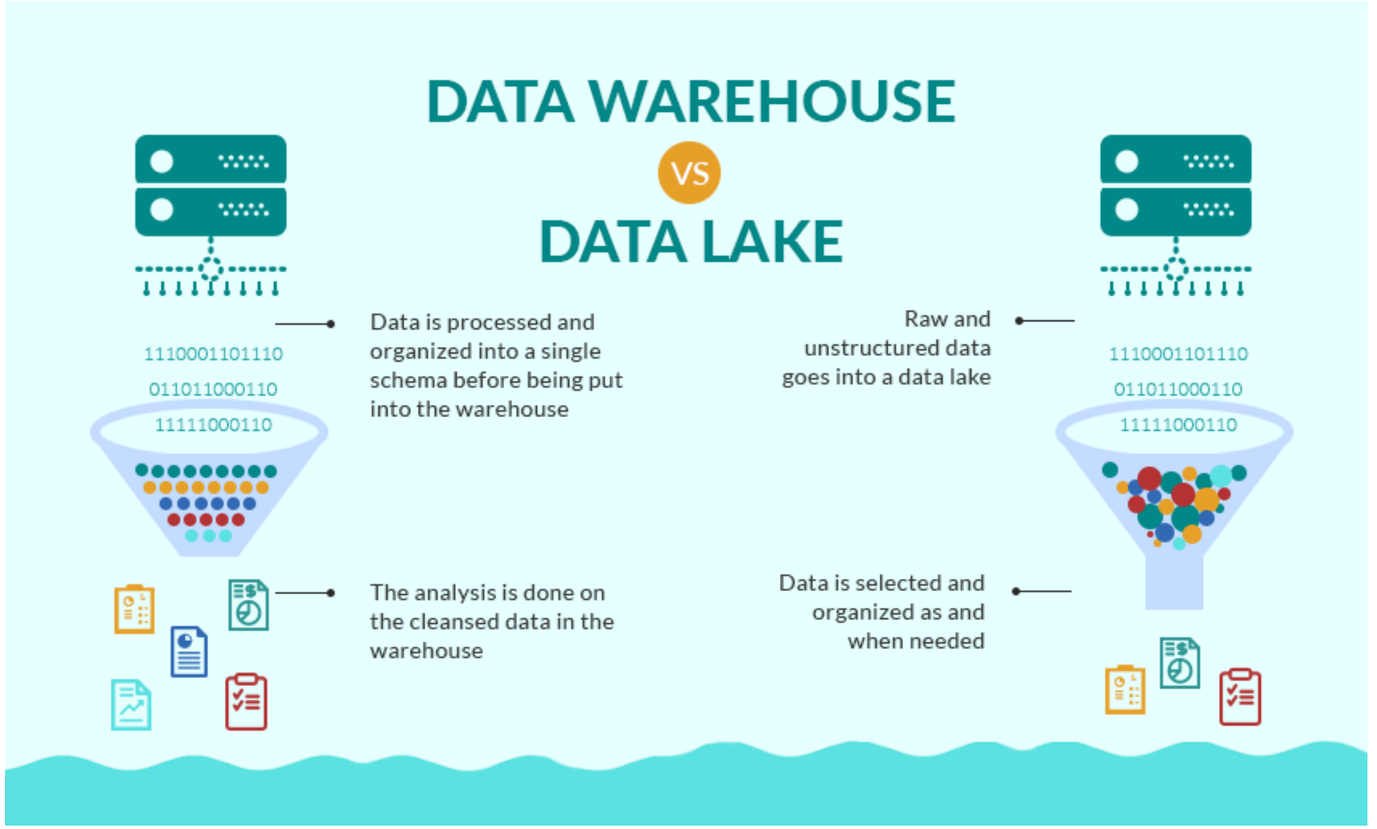
	Definición	**Concepto**
1	Para cada grupo de filas con los mismos CIds, aparece al menos un cierto número de veces repetidos.	kAn
2	Para cada grupo de filas con los mismos CIds, por cada atributo sensible S, existen un cierto número de valores distintos.	lDiv
3	Es el proceso que vuelve a asociar una persona con un conjunto de datos.	ReId
4	Es el proceso que remueve la asociación entre una persona y un conjunto de datos.	DeId
5	Son datos generados artificialmente, como forma opuesta a recolectarlos del mundo real.	DS

Data Science

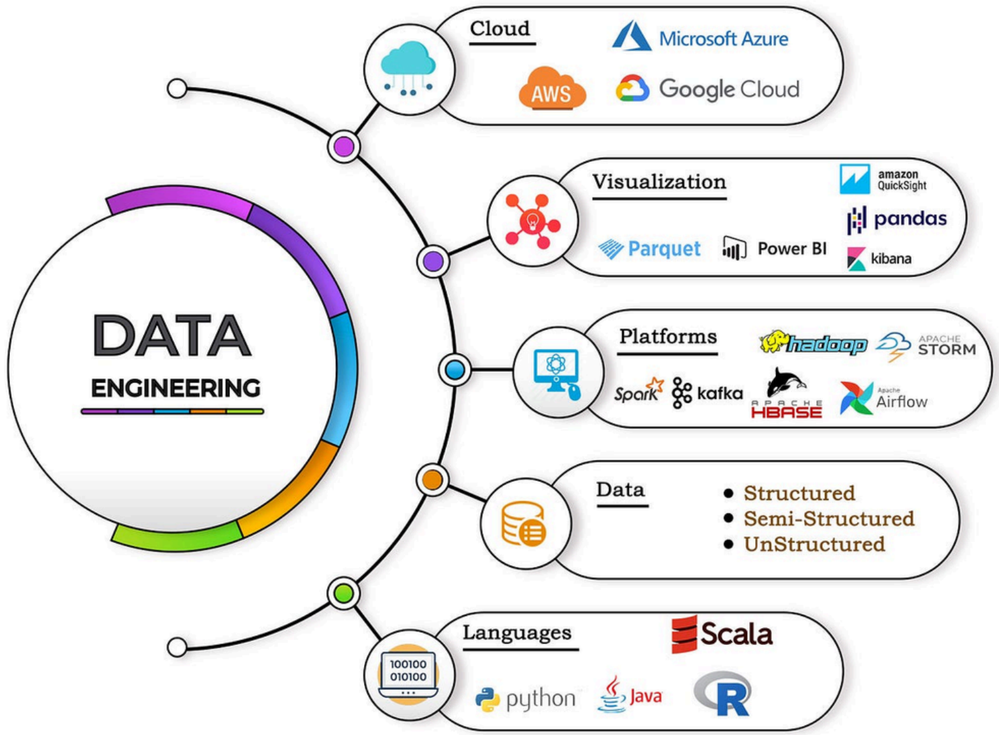


Data Engineering: Ejemplo de un sistema




DATA WAREHOUSE
VS
DATA LAKE
Data is processed and organized into a single schema before being put into the warehouse
1110001101110
011011000110
11111000110
The analysis is done on the cleansed data in the warehouse
Raw and unstructured data goes into a data lake
1110001101110
011011000110
11111000110
Data is selected and organized as and when needed

Data Engineer





MACHINE LEARNING

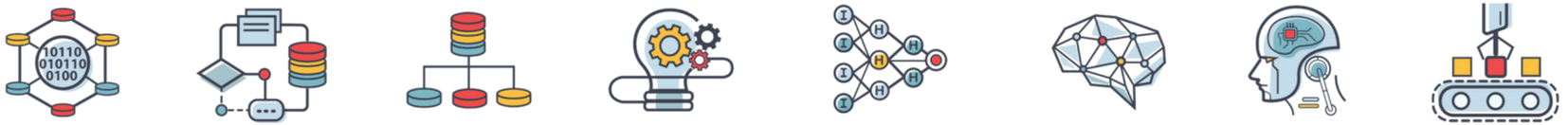


Tabla resumen del contenido

Tópico	Concepto Clave	Descripción y Detalles
Data Science	**Definición**	Enfoque multidisciplinario para extraer conocimientos prácticos de grandes volúmenes de datos, abarcando preparación, análisis y presentación de resultados.
	Áreas que engloba	Incluye Data Analysis, Machine Learning, Deep Learning, Computer Vision y Procesamiento de Lenguaje Natural (NLP).
	Proceso General	Se compone de: Recolección, Limpieza, Análisis Exploratorio (EDA), Construcción del Modelo y Despliegue del Modelo.
Librería Pandas	**Definición**	Herramienta de código abierto en Python para la manipulación y análisis de datos. Es rápida, potente y flexible.
	Estructura: DataFrame	Estructura de datos tabular bidimensional, mutable y potencialmente heterogénea, con ejes etiquetados (filas y columnas).
	Operaciones Clave	Merge: Uniones estilo base de datos (joins) .Query: Filtrado de columnas usando expresiones booleanas .GroupBy: Operaciones de dividir, aplicar función y combinar resultados.
Data Engineering	**Definición**	Rama de la ingeniería de software dedicada al desarrollo de infraestructura para transmisión, procesamiento, almacenamiento y acceso de datos.
	Objetivo	Proveer plataformas confiables y eficientes para que distintos consumidores (Data Scientists, BI, Developers) accedan a los datos 24/7.
	Pipeline de Datos	Flujo que va desde la Ingesta (fuentes), pasando por almacenamiento "Raw" (crudo), procesos ETL, hasta datos "Curados" (Data Lakes) y "Conformados" (Warehouses).
Almacenamiento	**Data Lake**	Repositorio que almacena copias de datos (a veces crudos) y mantiene información histórica. No suele actualizarse en tiempo real.
	Data Warehouse	Base de datos optimizada para consultas analíticas complejas (OLAP). Suele usar almacenamiento columnar y permite alta paralelización (ej. BigQuery, Redshift).
	ETL	Siglas de Extract, Transform, Load. Scripts periódicos que toman datos de fuentes, los procesan y los guardan en un destino final.
Herramientas	**Open Source**	Apache Spark: Procesamiento distribuido.Kafka: Streaming distribuido.Airflow: Orquestación de ETLs.
	Proveedores Cloud	Infraestructura gestionada para pipelines de Big Data ofrecida por Amazon (AWS), Microsoft (Azure) y Google (GCP).

¡MUCHAS GRACIAS!

IIC2413

AYUDANTÍA 13

REPASO EXAMEN