Curso

Data Engineer

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Centro de e-Learning UTN.BA

Módulo 1:

Extracción y almacenamiento de datos

Unidad 2:

Extracción de datos

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Presentación

La extracción de datos es la primera etapa fundamental en el ciclo de Ingeniería de datos. La extracción de datos es el punto de partida esencial para cualquier proyecto de ingeniería de datos, ya que proporciona los insumos necesarios para las etapas posteriores, como la transformación, limpieza, carga y análisis de datos.

En la extracción se recopilan datos de diversas fuentes para transferirlos a un entorno de trabajo adecuado para su posterior procesamiento y análisis. En esta unidad describiremos los distintos tipos de fuentes de datos que los/as Data Engineers suelen tener acceso y veremos cómo obtener los datos de cada una de ellas.

Además, analizaremos los formatos y las estructuras en las que suelen presentarse los datos. Y por último, detallaremos las técnicas de extracción más comunes utilizadas en Ingeniería de datos. Hablaremos de técnicas como la extracción incremental y la extracción “full” y explicaremos cuándo y por qué utilizar cada una.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bloques temáticos

1. Fuentes de datos
2. Formatos y estructuras
3. Técnicas

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Fuentes de datos

Si comienzas en un proyecto de Ingeniería de datos, una de las primeras tareas a realizar es la extracción de datos. Las organizaciones pueden contar con una variedad enorme de fuentes, u orígenes, de datos. Cada proceso dentro de una organización gestiona sus datos por medio de algún sistema. Dichos sistemas, expondrán alguna forma de obtener y extraer los datos deseados. A eso podemos considerar fuente de datos.

A continuación vamos a describir algunas de las fuentes de datos disponibles en una organización, con las que podremos trabajar como Data Engineer.

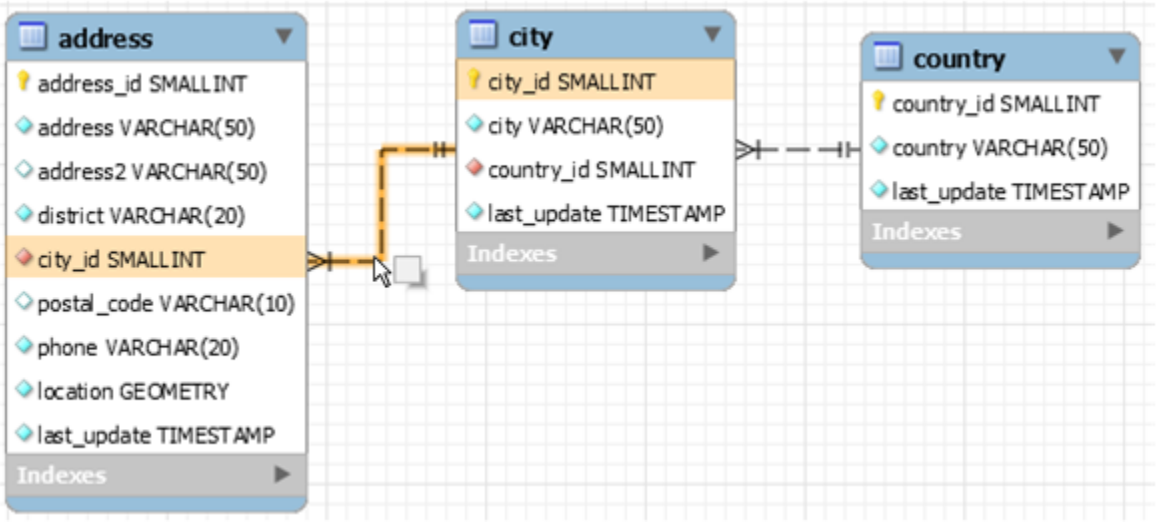
Bases de datos

Es muy probable que muchos procesos de una organización registren sus datos en una **base de datos relacional**. Este tipo de base de datos también recibe el nombre de **"transaccional"** y es caracterizada por ser un sistema **OLTP**. Las bases de datos transaccionales están diseñadas para recibir muchas escrituras, ya que los procesos de una organización suelen, con mucha frecuencia, crear o modificar datos. Este tipo de base de datos debe ser capaz de soportar esta carga de trabajo de forma óptima.

No se debe realizar analítica sobre las bases de datos OLTP ya que están optimizadas para operaciones de escritura. La analítica de datos realiza grandes operaciones de lectura sobre la base de datos, utilizando bases de datos o sistemas OLAP.

Las bases de datos relacionales organizan los datos por medio de tablas, que están compuestas por filas y columnas. Las tablas de una base de datos suelen estar conectadas entre sí por medio de relaciones a partir de columnas que tengan en común.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Ejemplo de un modelo relacional de base de datos

La base de datos es un concepto lógico para organizar y almacenar datos en un servidor. Son implementadas por medio de un software, conocido como motor de base de datos. Existen diferentes motores de base de datos: MySQL, SQL Server, PostgreSQL, etc. En el caso de bases de datos no relacionales, algunos motores son: MongoDB, Redis, DynamoDB, etc.

Como Data Engineer, vamos a interactuar con esos motores de base de datos y seremos los responsables de exportar los datos de las tablas y llevarlos hacia otro lugar por medio de un pipeline.

Para interactuar con la base de datos, por un lado, vamos a utilizar el lenguaje **SQL** para realizar consultas. Por otro lado, haremos uso de alguna herramienta **ETL** o de un lenguaje de programación para extraer los datos de las tablas. Cabe aclarar que estos procesos son posibles por medio de algún protocolo,

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

como JDBC u ODBC, los cuales van a requerir la instalación de drivers específicos al motor de base de datos.

APIs

Como fuente de datos, una API actúa como un punto de acceso que permite obtener datos de una aplicación, servicio o sistema externo, sin tener que acceder directamente a la base de datos subyacente.

Una API proporciona una interfaz estructurada y programática para acceder a los datos almacenados en una aplicación o servicio. Puede ofrecer una forma estandarizada de solicitar datos específicos utilizando métodos de consulta y parámetros predefinidos.

Además, las APIs suelen proporcionar diferentes “endpoints”, que representan distintos conjuntos de datos o funcionalidades específicas, como filtrar.

Las APIs usan el HTTP, un protocolo de comunicación utilizado en la web para la transferencia de datos. Está compuesto por una serie de métodos (GET, POST, PUT, DELETE, entre otros) que permiten realizar diferentes operaciones en los recursos a través de las URLs. Por ejemplo, el método GET se utiliza para solicitar datos, el método POST se utiliza para enviar datos, etc.

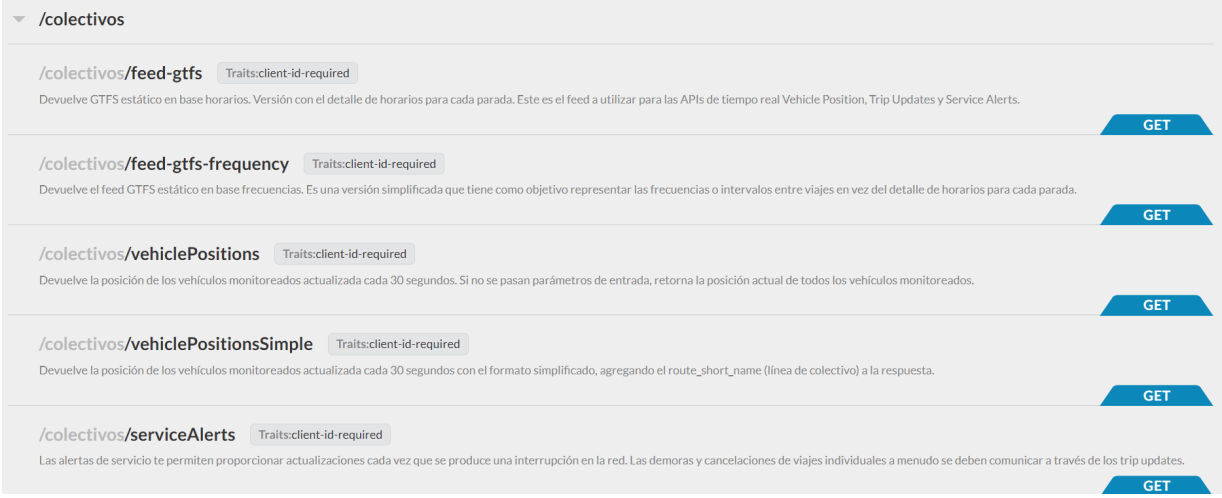
Al realizar una solicitud a una API, devuelve una respuesta en formatos JSON o XML, por lo general.

Por último, las herramientas a utilizar para extraer datos de una API pueden ser:

- una plataforma ETL o de Integración, como Apache Nifi, que ofrecen componentes predefinidos para conectarse y extraer datos de APIs.
- algún lenguaje de programación, como Python, junto con alguna librería para realizar solicitudes HTTP a la API.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

- Otras aplicaciones como Postman (basada en una interfaz gráfica) y cURL (basada en consola), que son útiles para pruebas sencillas.



Ejemplo de una API pública que brinda datos abiertos sobre el transporte en CABA

Archivos

En algunas ocasiones, los datos a obtener estarán disponibles en formato de archivos. Nos vamos a encontrar con diferentes formatos: Imágenes en formato PNG o JPG, archivos PDFs, CSV, TXT, Excels u hojas de cálculo (cabe aclarar que las hojas de cálculo suelen abundar mucho en una organización), entre otros.

Estos archivos pueden estar disponibles en un servidor SFTP o en algún repositorio en la nube como OneDrive, Sharepoint, Google Drive, etc.

Además del contenido de los archivos, son muy importantes sus metadatos. Se trata de datos acerca de los datos, que se caracterizan por describir el

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

contenido, calidad, condiciones, historia, disponibilidad y otras características de los datos.

Por último, la ingesta de archivos se realizará por medio de alguna herramienta, que se encargue de mover los archivos desde sus fuentes hasta el destino deseado, como un data lake. Luego se utilizará alguna herramienta capaz de extraer su contenido y darle un formato más adecuado para el análisis.

Formatos y estructuras

Aparte de existir diversas fuentes que proveen datos, existen diferentes estructuras y formatos en las que se presentan.

Vamos a hablar sobre datos estructurados, semi-estructurados y no estructurados, junto con los formatos más comunes en cada uno de ellos.

Datos estructurados

Son datos organizados en una estructura definida y uniforme, generalmente en forma de tablas con filas y columnas. Cada columna tiene un nombre y un tipo de dato predefinido. Los datos estructurados siguen un esquema fijo y están altamente organizados.

Los datos estructurados los encontraremos, por ejemplo, en las tablas de una base de datos relacional.

Otro formato común en los datos estructurados son los archivos CSV. Es un formato de texto plano, donde cada línea representa una fila de datos, y los valores de cada columna están separados por un carácter delimitador, como una coma, punto y coma, tabulación, etc. Es especialmente adecuado para representar datos tabulares.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

	A	B	C	D	E	F	G	H	I
1	Sales Representative	Location	Region	Customer	Order Date	Item	Quantity	Price	Total Sale Amount
2	Sara Snyder	New York	East	Phyllis Johnston	2016-10-30	Things	1	17.83	17.83
3	Sara Snyder	New York	East	Kimberly Little	2016-05-23	Junk	3	12.42	37.26
4	Frances Warren	Massachusetts	East	Justin Dixon	2016-09-27	Widgets	4	53.35	213.4
5	Sara Snyder	Massachusetts	East	Shirley Rivera	2016-02-12	Junk	5	12.42	62.1
6	Diane Gonzalez	Oregon	West	Marilyn Franklin	2016-02-14	Things	8	17.83	142.64
7	Patrick Graham	Washington	West	Henry Sanders	2016-04-11	Widgets	4	53.35	213.4
8	Sara Snyder	Connecticut	East	Benjamin Phillips	2016-09-02	Junk	4	12.42	49.68
9	Frances Warren	New Jersey	East	Theresa Torres	2016-11-26	Junk	4	12.42	49.68
10	Patrick Graham	Oregon	West	Roger Bell	2016-07-13	Junk	10	12.42	124.2
11	Sara Snyder	New Jersey	East	Harold Matthews	2016-06-02	Junk	3	12.42	37.26
12	Frances Warren	New York	East	Roy Young	2016-06-02	Widgets	8	53.35	426.8
13	Sara Snyder	New York	East	Debra Allen	2016-02-20	Things	1	17.83	17.83
14	Randy Watson	Connecticut	East	Alan Dean	2016-06-07	Junk	7	12.42	86.94
15	Randy Watson	Massachusetts	East	Robin Matthews	2016-10-31	Stuff	5	16.32	81.6
16	Randy Watson	New York	East	Randy Burton	2016-03-13	Stuff	4	16.32	65.28

Ejemplo de datos en formato tabular

Datos semi-estructurados

Se trata de esquemas menos rígidos y más flexibles que los datos estructurados. No requieren un esquema predefinido, lo que los hace más flexibles y adaptables a cambios en la estructura o el formato de los datos.

Por un lado, este tipo de datos se almacenan en bases de datos no relaciones, o NoSQL (Not Only SQL). Por otro lado, los formatos más comunes aquí son el JSON, XML, YML, etc.

JSON representa datos en forma de pares clave-valor, es muy utilizado en el intercambio de datos, por eso predomina como formato de salida en la respuesta de una API.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

JSON

```json
{
    "firstName": "Jane",
    "lastName": "Doe",
    "hobbies": ["running", "sky diving", "singing"],
    "age": 35,
    "children": [
        {
            "firstName": "Alice",
            "age": 6
        },
        {
            "firstName": "Bob",
            "age": 8
        }
    ]
}
```

Datos en formato JSON

Datos no estructurados

Los datos no estructurados son datos que no siguen un modelo y no pueden ser representandos en un formato de filas y columnas. Esto dificulta su búsqueda y organización. Suelen ser texto, sonido, imágenes o vídeos.

Este tipo de datos pueden ser extremadamente valiosos, pero como son difíciles de buscar y organizar, este valor no se ha podido extraer hasta hace poco, con la llegada del machine learning y la inteligencia artificial.

Por ejemplo, Computer Vision se encarga de explotar el contenido de imágenes y vídeos, mientras en el procesamiento del lenguaje natural procesa textos sin formato disponible en documentos, mails, etc.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Técnicas de extracción de datos

Una ingesta o recolección de datos se puede realizar de dos formas:

- De forma batch
- O en tiempo real

Ingesta batch

La ingesta de datos de tipo batch es aquella que opera sobre un volumen de datos definido, también conocido como lote. Su ejecución se realiza durante un periodo de tiempo y de forma periódica (ya sea cada 10 diez minutos, cada día, cada semana, etc.)

Aquí nos podemos podemos encontrar con dos tipos de ingestas:

Ingesta full

Se caracteriza por extraer todos los datos de la fuente en cada ejecución del proceso de extracción y volcarlos por completo en el sistema de destino. El volcado de los datos puede sobreescribir lo que ya está disponible en el sistema de destino, o depositar los datos en otro archivo o directorio. Puede ser útil cuando se trabaja con fuentes de datos estáticas o cuando los requisitos del proyecto no permiten la identificación de cambios incrementales.

Para este tipo de ingesta hay que tener en cuenta el volúmen de datos, ya que en cada ejecución se toma una captura completa de la tabla o fuente de datos de orígen.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Ingesta incremental (o delta)

Consiste en recolectar actualizaciones de la fuente de datos, ya sea por la inserción de nuevos registros o la modificación de los existentes. En lugar de extraer todos los datos, se realizan consultas que seleccionan solo los datos nuevos o modificados desde la última extracción.

El rastreo de cambios en esta técnica es posible si la fuente de datos cuenta con algún campo de identificación o con alguna marca de tiempo.

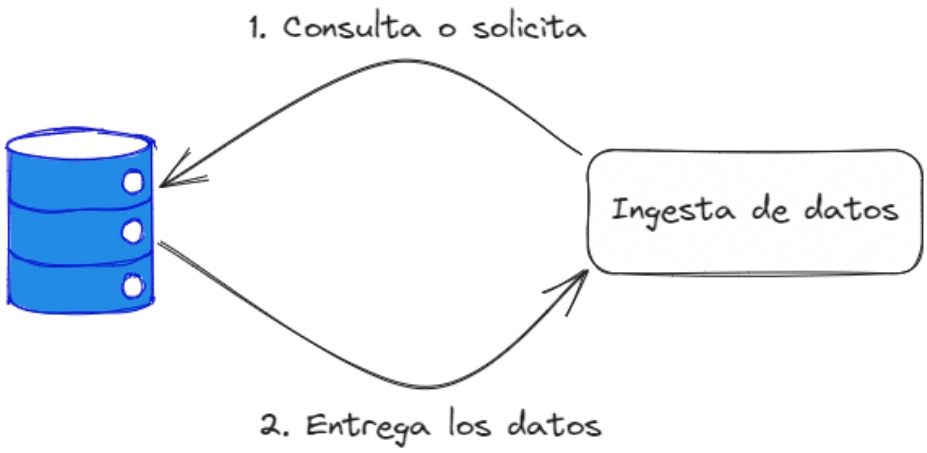
Ingesta en tiempo real

Implica la captura de datos en tiempo real, a medida que se generan. En lugar de esperar a que se realice una extracción programada, los datos se recolectan de inmediato, lo que permite obtener información actualizada en tiempo real.

La ingesta batch se caracteriza por su ejecución programada de forma periódica, cada ejecución consulta a la fuente y obtiene un lote de datos. En la ingesta en tiempo real, los datos fluyen desde la fuente hacia el sistema destino, no existe una consulta o petición por parte del recolector de datos.

Como ejemplo de datos en tiempo real, nos podemos encontrar con las mediciones de sensores o con publicaciones e interacciones de una red social. Dichos datos suelen ser denominados como mensajes o eventos. En Ingeniería de datos, los eventos o mensajes deben arribar a una “cola de mensajería” (message queue) o a una “plataforma de streaming de datos”. Este tipo de tecnología se encarga de ordenar los datos recibidos y de asegurar que sean procesados una sola vez, entre cosas que detallaremos en otra unidad. Algunas tecnologías que permiten esto son Apache Kafka, Apache Pulsar, MQTT, RabbitMQ, etc.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Flujo de datos en ingesta batch

Flujo de datos en ingesta en tiempo real

Change Data Capture

Por último, existen ocasiones donde se necesita capturar cambios de una base de datos en tiempo real. CDC (Change Data Capture) es una técnica utilizada para identificar y capturar los cambios realizados en una base de datos en

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

tiempo real. El objetivo principal es detectar los cambios de manera eficiente y registrarlos para su procesamiento posterior.

CDC permite capturar los cambios realizados en una fuente de datos sin tener que recorrer todos los datos nuevamente. En lugar de extraer todos los registros de una tabla, CDC registra solo los cambios realizados, como inserciones, actualizaciones o eliminaciones de registros individuales.

Existen diferentes formas de implementar CDC. Una se basa en un log de transacciones. En los motores de bases de datos, el log de transacciones registra las operaciones realizadas en la base de datos en un orden secuencial. CDC puede leer el log y capturar los cambios registrados en él.

Para implementar este tipo de CDC se utilizan las tecnologías de Debezium y Apache Kafka. Debezium está diseñado para leer los logs de transacciones de base de datos y convertir los cambios en eventos estructurados. Estos eventos se envían a Apache Kafka, para que los datos estén disponibles para su consumo casi instantáneamente.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Conclusión

En conclusión, la unidad temática 2 nos ha brindado una comprensión sólida sobre la extracción de datos en el campo de la ingeniería de datos. Hemos explorado las diversas fuentes de datos disponibles, que incluyen bases de datos, APIs y archivos.

Además, hemos profundizado en los diferentes formatos y estructuras de datos que encontraremos en estas fuentes. Desde datos estructurados en bases de datos relacionales, semi-estructurados en archivos JSON y no estructurados en imágenes y texto, hemos aprendido a reconocer y trabajar con diferentes tipos de formatos. Comprender las particularidades de cada formato es esencial para realizar la extracción y transformación de datos de manera efectiva.

Hemos examinado las distintas formas de extraer datos, tanto en lotes (batch) como en tiempo real. Además, hemos abordado la diferencia entre extracción completa e incremental de datos. La extracción completa implica recuperar todos los datos disponibles en una fuente en cada ejecución, lo que puede ser apropiado cuando se requiere una imagen completa y actualizada de los datos. Por otro lado, la extracción incremental se centra en identificar y extraer solo los datos nuevos o modificados desde la última ejecución, lo que puede ser más eficiente y útil para actualizaciones periódicas o continuas.

Aplicaremos todo lo aprendido por medio de alguna herramienta como Python y continuaremos profundizando en las otras responsabilidades del/la Data Engineer en las siguientes unidades.

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bibliografía utilizada y sugerida

- Explore core data concepts - Training. (s. f.). Microsoft Learn. https://learn.microsoft.com/en-us/training/modules/explore-core-data-concepts/

- Explore fundamental relational data concepts - Training. (s. f.). Microsoft Learn. https://learn.microsoft.com/en-us/training/modules/explore-relational-data-offerings/

- Change Data Capture. (s. f.). Data Engineering Wiki. https://dataengineering.wiki/Concepts/Change+Data+Capture

- Delta Load. (s. f.). Data Engineering Wiki. https://dataengineering.wiki/Concepts/Delta+Load

- Full Load. (s. f.). Data Engineering Wiki. https://dataengineering.wiki/Concepts/Full+Load

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning