Unidad 2. Machine Learning para la Ciencia de Datos

SQL y Adquisición de Datos

CODERHOUSE

Introducción a la Adquisición de datos

CODERHOUSE

Comprensión del problema de negocio

¿Cómo podemos empezar?

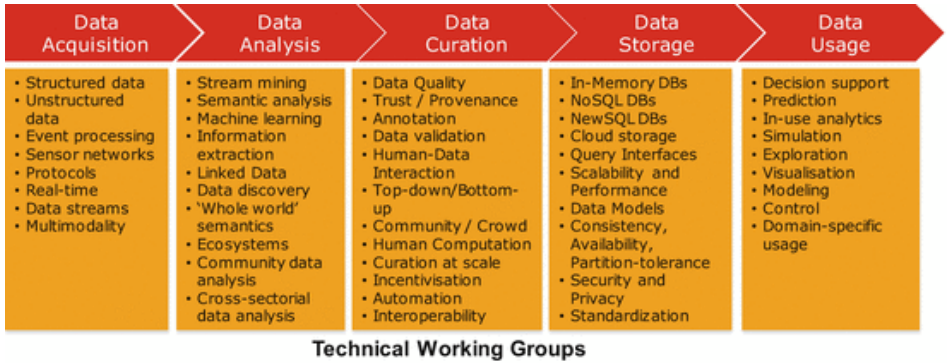
Punto de partida

- ✓ ¿Qué problema debemos solucionar?
- ✓ ¿Qué tipo de datos se requieren para hacer el análisis?
- ✓ ¿Dónde podemos encontrar dichos datos?
- ✓ ¿Cómo puedo acceder a los datos?
- ✓ ¿Los datos que deseamos realmente existen?

Contesta en el chat de Zoom

Big Data Value Chain

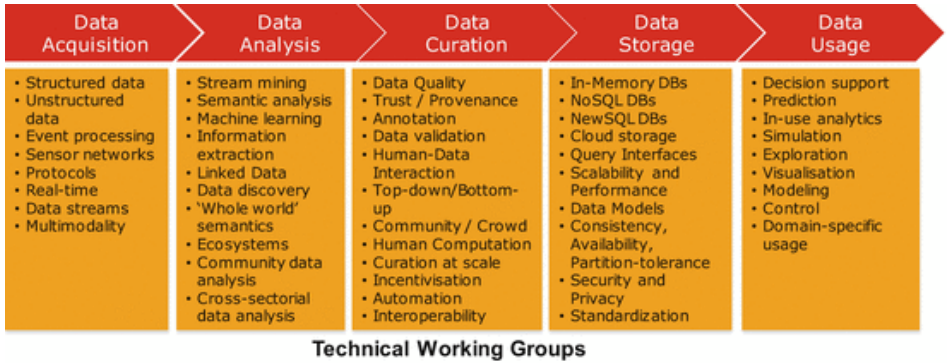
Big Data Value Chain



- ✓ **Data acquisition:** es el proceso de recopilación, filtrado y limpieza de datos antes de colocarlos en un almacén de datos o cualquier otra solución de almacenamiento en la que se pueda realizar el análisis de datos.
- ✓ **Data analysis:** hacer que los datos sin procesar adquiridos sean aptos para su uso en la toma de decisiones
- ✓ **Data Curation:** gestión activa de los datos durante su ciclo de vida para garantizar que cumplan con los requisitos de calidad de datos necesarios para su uso efectivo

Fuente: The Big Data Value Chain: Definitions, Concepts, and Theoretical Approaches

Big Data Value Chain



- ✓ **Data Storage:** es la persistencia y gestión de datos de forma escalable que satisface las necesidades de las aplicaciones que requieren un acceso rápido a los datos
- ✓ **Data Usage:** actividades comerciales basadas en datos que necesitan acceso a los datos, su análisis y las herramientas necesarias para integrar el análisis de datos dentro de la actividad comercial.

Fuente: The Big Data Value Chain: Definitions, Concepts, and Theoretical Approaches

Lectura de fuentes de datos con Pandas

Lectura de datos con Pandas

Recordemos algunas de las características de pandas:

- ✓ Nos permite lidiar con archivos con codificaciones raras (parámetro encoding)
- ✓ Nos permite manipular encabezados y columnas de archivos
- ✓ Permite manipulación y estructuración de datos en formato fecha
- ✓ Definir tipos de datos a priori en la lectura
- ✓ Identificar instancias inválidas
- ✓ Concatenar y manipular archivos

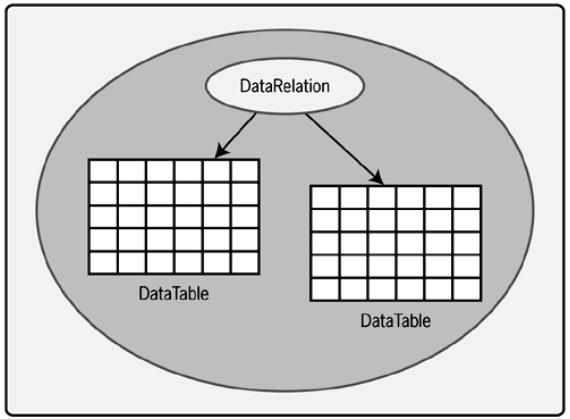
Headers, Booleanos y Fechas

Lectura de archivos planos (Headers)

test.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
a b c d message
1 2 3 4 hello
5 6 7 8 world
9 10 11 12 foo
```




```
pd.read_csv('test.txt',sep=' ', header = None )
```

	0	1	2	3	4
0	a	b	c	d	message
1	1	2	3	4	hello
2	5	6	7	8	world
3	9	10	11	12	foo

```
In [7]: pd.read_csv('test.txt',sep=' ',  names=['z', 'u', 't', 'y', 'mensaje'] )
```

Out[7]:

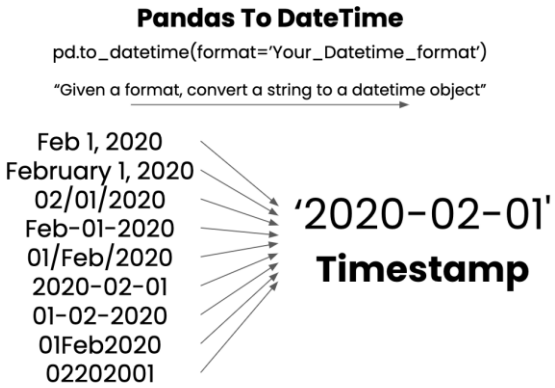
	z	u	t	y	mensaje
0	a	b	c	d	message
1	1	2	3	4	hello
2	5	6	7	8	world
3	9	10	11	12	foo

CODERHOUSE

Lectura de archivos planos (Fechas)

Parseo automático de fechas

```
date,product,price
1/1/2019,A,10
1/2/2020,B,20
1/3/1998,C,30
```




```python
import pandas as pd
```

```python
df = pd.read_csv('dates.txt',sep=',')
df.dtypes
```

```
date       object
product    object
price       int64
dtype: object
```

```python
df = pd.read_csv('dates.txt',sep=',',parse_dates=['date'])
df.dtypes
```

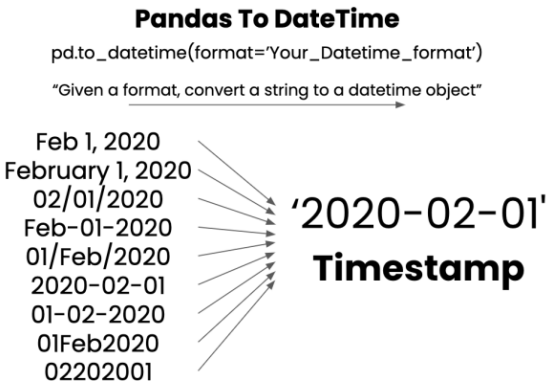
```
date       datetime64[ns]
product            object
price               int64
dtype: object
```

CODERHOUSE

Lectura de archivos planos (Fechas)

Parsing manual de fechas

```
year,month,day,product,price
2019,1,1,A,10
2019,1,2,B,20
2019,1,3,C,30
2019,1,4,D,40
```



👉

```
df = pd.read_csv('data/data_4.csv',
                 parse_dates=[['year', 'month', 'day']])
df.info()

RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   year_month_day  4 non-null      datetime64[ns]
 1   product         4 non-null      object
 2   price           4 non-null      int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 224.0+ bytes
```

CODERHOUSE

Lectura de archivos planos (Booleanos)

Read_csv puede detectar automáticamente booleanos si se le indica.
"asistió" se refiere a la asistencia de un alumno y "Tarea" si completo la tarea o no.

```
ID,asistio,Tarea
cef36,1,1.0
323e5,0,0.0
b29a1,0,0.0
04a11,1,0.0
936s2,1,1.0
dd0e7,0,0.0
```

```
pd.DataFrame ({'float' : [2.0],
               'int' : [2],
               'datetime' : [pd.Timestamp ('20190210')],
               'string' : ['f1']})
```


```
1  df = pd.read_csv('Ejemplo_encuesta.txt', dtype ={"asistio":bool,
2                                                   "Tarea":bool})
3  df
```

	ID	asistio	Tarea
0	cef36	True	1.0
1	323e5	False	0.0
2	b29a1	False	0.0
3	04a11	True	0.0
4	936s2	True	1.0
5	dd0e7	False	0.0

Hojas de cálculo

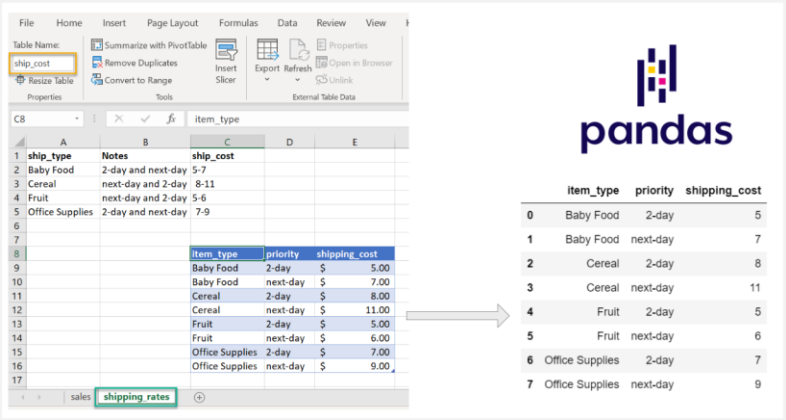
Hojas de cálculo

- ✓ Son datos almacenados de forma tabular en filas y columnas
- ✓ Cada fila se considera una instancia y cada columna una variable
- ✓ A diferencia de los archivos en formato plano, pueden tener fórmulas y formato
- ✓ Un solo archivo puede tener varias hojas de cálculo
- ✓ Tienen algunas limitaciones de cantidad de almacenamiento y velocidad de procesamiento

Hojas de cálculo

```
In [2]: pd.read_excel?
```

```
Signature:
pd.read_excel(
    io,
    sheet_name=0,
    header=0,
    names=None,
    index_col=None,
    usecols=None,
    squeeze=False,
    dtype=None,
    engine=None,
    converters=None,
    true_values=None,
    false_values=None,
    skiprows=None,
    nrows=None,
    na_values=None,
    keep_default_na=True,
    na_filter=True,
    verbose=False,
```



La lectura de hojas de cálculo permite muchas opciones una de las más importantes es la lectura de hojas específicas, eliminación de columnas indeseables e índices innecesarios

CODERHOUSE

Hojas de cálculo

Lectura de hojas y columnas

```
#Seleccionando columnas y hoka
df = pd.read_excel(file_location,sheet_name='Sheet1', usecols="A,C,F")
```

```
# Seleecionando numero de hoja y rango de columna
df = pd.read_excel(file_location,sheet_name=1, usecols="A:N")
```

```
# Rango de columnas
df = pd.read_excel(file_location,sheet_name='Sheet2', usecols="A:F,H")
```

```
# Rangos de columnas
df = pd.read_excel(file_location,sheet_name='Sheet1', usecols="A:F,H,J:N")
```

Hojas de cálculo

Podemos, en lugar de usar el método pd.read_excel, crear un objeto y hacer un parsing las hojas del excel.

```
xls = pd.ExcelFile(file_location)
xls.parse(sheet_name='Clase 1')
```

```
pd.read_excel(file_location, sheet_name ='Clase 1')
```

```
1 xls.sheet_names
```

```
['Clase 1',
 'Clase 2',
 'Clase 3',
 'Clase 4',
 'Clase 5',
 'Clase 6',
 'Clase 7',
 'Clase 8',
 'Clase 9',
 'Clase 10',
 'Clase 11',
 'Clase 12',
 'Clase 13',
 'Clase 14',
 'Clase 15',
 'Clase 16',
 'Clase 17',
 'Clase 18',
```

Atributo que enumera los nombres de las hojas.

Otros formatos estructurados (Pickle)

Es un formato nativo de python que es popular para la serialización de objetos.

- ✓ Es mucho más rápido que .csv.
- ✓ Reduce el tamaño de los archivos en la mitad usando técnicas de compresión.
- ✓ No hay necesidad de especificar columnas de datos ni argumentos.

- ✓ Al ser nativo de python solo puede leerse utilizando python.

ACTIVIDAD EN CLASE

Lectura de datos con Pandas

Aprenderemos a manipular SQL, JSON y APIs.

Exploramos las librerías sqlite3, yfinance y la función read_json con el fin de comprender cómo leer y procesar archivos en diferentes formatos.

1. Leer las tablas NBA_season1718_salary y Season_Stats dentro del archivo nba_salary.sqlite
2. Leer el archivo JSON en la siguiente página Web: JSON file
3. Utilizar la funcion Ticker de yfinance para extraer información relevante de la compañias PFE (Pfizer)

Trabajaremos de forma **individual.** Se estiman **5 minutos para cada ejercicio de lectura y 5 min para compartir las conclusiones.**

Structured Query Language (SQL)

Sublenguajes (SQL)

DDL Conjunto de sentencias para la **definición y modificación** de la base de datos y sus tablas.

DCL Conjunto de sentencias para la **administración de los privilegios** de los distintos usuarios que se conectarán a la base de datos.

DML Conjunto de sentencias para la **manipulación de los datos** almacenados.

TCL Conjunto de sentencias para la para la **gestión de transacciones**.

Para esta clase...

A lo largo del curso **se mostrarán ejemplos utilizando la base de datos de W3School** provista en este enlace.

Si bien la herramienta cuenta con varias limitaciones en comparación con un sistema de gestión de bases de datos real, **será suficiente para permitirnos un primer acercamiento a SQL**.

Lenguaje de definición de datos (DDL)

Sentencias básicas de DDL

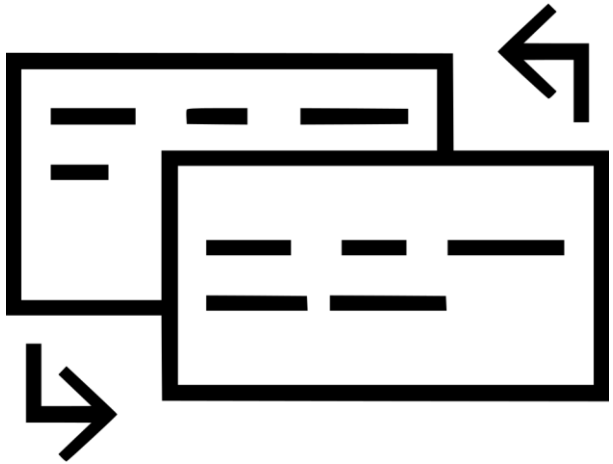
- CREATE
- ALTER
- DROP
- TRUNCATE

Mediante este tipo de **operaciones** es posible **definir nuevas tablas, modificarlas, eliminarlas o vaciarlas.**

CREATE

La creación de la base de datos y su esquema puede realizarse de dos maneras:

1. Utilizando la interfaz gráfica del **Sistema Gestor de Base de Datos** (SGBD)
2. Por medio de la sentencia **CREATE SCHEMA**.



DROP

- Se usa para eliminar una **base de datos** completa o solo una **tabla**.

- Destruye los objetos como una **base de datos, tabla, índice o vista** existente.

- También se pueden remover índices, triggers, constantes y permisos

ALTER TABLE

ALTER TABLE se usa para agregar, eliminar o modificar columnas en una tabla existente. También se usa para agregar y eliminar varias restricciones en una tabla existente.

TRUNCATE

1. TRUNCATE TABLE se usa para eliminar datos completos de una tabla existente.

1. Permite eliminar la tabla completa, pero eliminaría la estructura de la tabla completa de la base de datos y se necesitaría volver a crear esta tabla una vez más si desea almacenar datos.

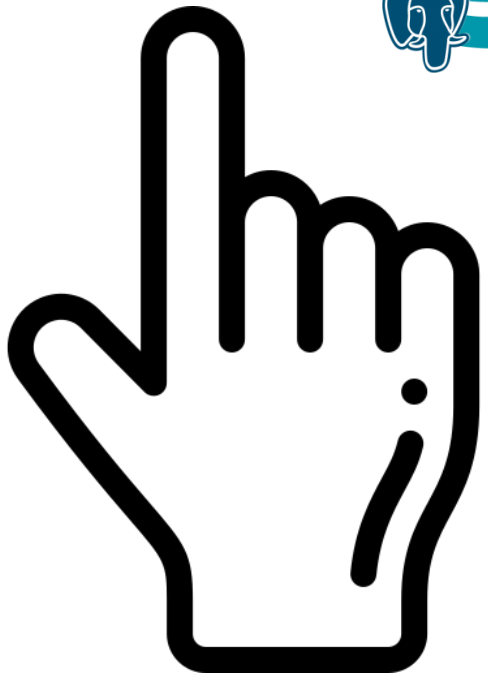
PARA RECORDAR

La sentencia SELECT FROM tiene la particularidad de que **no produce ningún cambio** en el estado de la base de datos. En cambio, **las tres restantes sí tienen la capacidad de producir algún cambio** en los registros almacenados, por lo que se deberá prestar especial atención.

SELECT

1. La instrucción SELECT se utiliza para seleccionar datos de una base de datos.

1. Los datos devueltos se almacenan en una tabla de resultados, denominada conjunto de resultados.



CODERHOUSE

Predicados simples

Sentencias básicas de DML

BETWEEN	Retorna verdadero si los valores de una columna están **entre los dos valores** dados.
IN	Retorna verdadero si los valores de una determinada columna **están en un determinado conjunto de valores**.
IS NULL	Retorna verdadero si el **campo está vacío**.
LIKE	Retorna verdadero si un determinado campo de tipo CHAR o VARCHAR **cumple con un patrón determinado.**

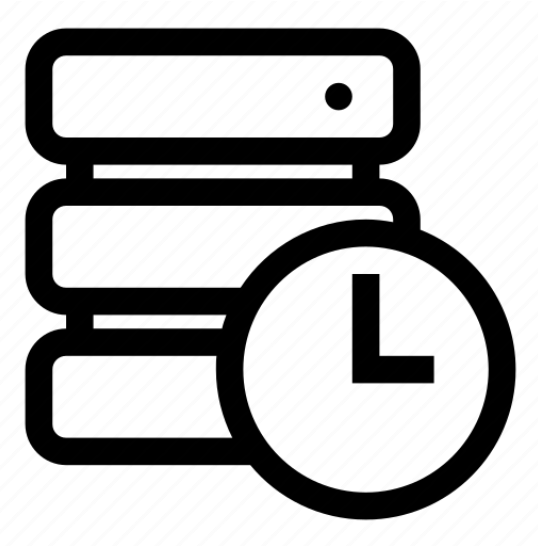
INSERT

La sentencia INSERT INTO se utiliza para insertar nuevos registros en una tabla. Es posible escribir la declaración INSERT INTO de dos maneras:

1. Especificar tanto los nombres de las columnas como los valores que se insertarán
2. Si está agregando valores para todas las columnas de la tabla, no necesita especificar los nombres de las columnas en la consulta SQL. Sin embargo, asegúrese de que el orden de los valores sea el mismo que el de las columnas de la tabla.

UPDATE

1. La instrucción UPDATE se utiliza para modificar los registros existentes en una tabla.
2. ¡Tengan cuidado al actualizar registros en una tabla!. La cláusula WHERE especifica qué registros deben actualizarse. Si omite la cláusula WHERE, ¡se actualizarán todos los registros de la tabla!



DELETE

1. La declaración DELETE se usa para eliminar registros existentes en una tabla.

1. ¡Tengan cuidado al eliminar registros en una tabla! Observe la cláusula WHERE en la instrucción DELETE. La cláusula WHERE especifica qué registros deben eliminarse. Si omite la cláusula WHERE, ¡se eliminarán todos los registros de la tabla!

Uso del lenguaje DML

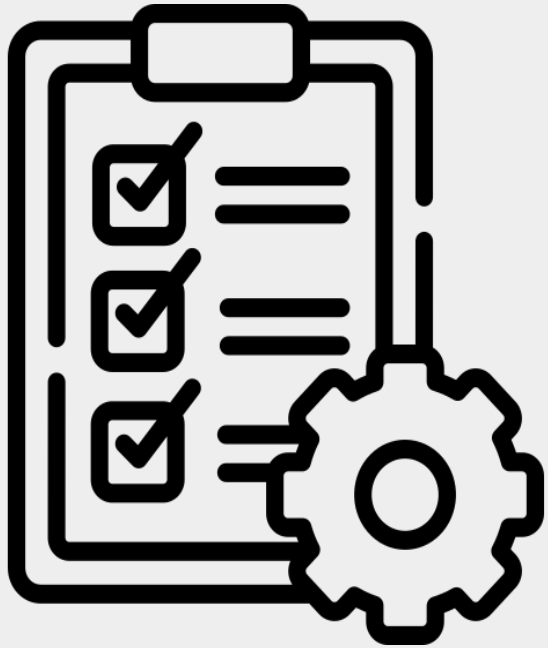
Duración: **15 minutos**

ACTIVIDAD EN CLASE

Uso del lenguaje DML

Utilizaremos la base de datos en este link

1. Identificar a los clientes de la tabla **Customers** que viven en París o Londres (8 registros)
2. Encontrar en la tabla **OrderDetails** las órdenes cuya cantidad está entre 10 y 15 y además que el OrderDetailID sea mayor que 6. (121 registros)
3. Seleccionar todos los registros de la tabla **Products** donde el ProductName empiezan por la letra C (9 registros)



CODERHOUSE

#CoderAlert

Encontrarás en la “Guía de actividades hacia el PF” del curso una actividad para aplicar todo lo aprendido hoy sobre Data Acquisition a tu Proyecto. ¡Será fundamental al momento de realizar tu primera Entrega N° 1 en Unidad 9!

Descarga de datos desde APIs públicas

Crearás un notebook donde se seleccionará una API de interés, luego crearás una API key y finalmente extraerás la información para ser almacenada en un DataFrame

CLASE N°1

Glosario

Pregunta problema: pregunta que resume el objetivo que se quiere resolver.

Big Data Value Chain: proceso para describir el flujo de información dentro de un sistema de big data para generar valor e información útil a partir de los datos.

Adquisición de datos: proceso de carga de datos para la resolución de un problema de interés.

Data Management Maturity Model: proporciona pautas para ayudar a las organizaciones a construir, mejorar y medir su capacidad de gestión de datos empresariales.

Tipos de datos: pueden ser estructurados, semi-estructurados y no estructurados.

APIs: un conjunto de funciones y procedimientos para la creación de aplicaciones que acceden a las características o datos de un sistema operativo u aplicación.

CLASE N°3

Glosario

DML: lenguaje que permite a los usuarios manipular datos en una base de datos (insertar, recuperar, eliminar y modificar datos existentes).

CREATE: se utiliza para crear una nueva tabla en una base de datos. Los parámetros de columna especifican los nombres de las columnas de la tabla.

DROP: se usa para eliminar una tabla existente en una base de datos.

DDL: lenguaje para crear y modificar la estructura de los objetos de base de datos en una base de datos (vistas, esquemas, tablas, índices)

SELECT: selecciona datos de una base de datos. Los datos obtenidos se almacenan en una tabla de resultados (conjunto de resultados).

INSERT: La sentencia INSERT INTO se utiliza para insertar nuevos registros en una tabla de una base de datos.