Curso Python

LinkedIn Learning

STARG VIDEK

Data Science Foundations:

Python Scientific Stack

OBJETIVOS: - Jupiter - Pandas - Folium
- NumPy - Conda - Geo
- ML con Sickit-Learn
- Plotting con Matplotlib y Bokeh
- Numba Cython, deep learning and NLP.
- virtual env. Kivy. Selenium

- NumPy → matrices. cálculo Numérico
  SciPy: colección con scikit y Science capabilities.
- Matplotlib → Visualizations.
- Pandas → dataset.
- Sickit-Learn → ML Algorithms
- Jupyter → ? → Ayuda
  ?? → ver código fuente
  %-pwd → ver directory

STARG VIDEK

%time → mide el tiempo de ejecutar un comando

!dir !ls. $ Python -V

cmd comandos en jupiter.

Podes usar Markdown cells y se ven dere-jo como HTMLS. En vez de usar comments normales de python. podes usar latex, img y más!

- Numpy

- NO ES la mejor opción en cuanto a uso de memoria y performance

```
array = np.array([1,2,3])
```

array.dtype → me devuelve el tipo de variable de mi array

Podes usar operadores logicos y Slicing Adentro de un Matriz!

```
array(1:,2:)  _and.
array((array>7) & (array<9))
```

array([1,2,3],[4,5,6]) → matriz

Boolean indexing (lo de fin pag anterior) es mucho más rapido y eficiente q usar FOR LOOPS!

dir(vector) → me muestra todo los atributos q tiene el dicho.

numpy espera ndarrays en vez de listas pq trig.

@.np.vectorize(function)

decorator de numpy

PANDAS

read_csv(FileName, parse_dates=['time'])

df['name color'] → ACCEDES COLS.

df['LAT'][0]

col fila.

df.loc[0] o df.loc[2:7]

devuelve toda la fila de la 2 a la 7.

df.[['LAT', 'long'][2:7]]

OBTENER filas 2:7 pero todas de esas dos cols.

df.LOC[Label Name]

↳ Puede ser una serie.

	Col 1	Col N
Index 1	data	data
:		
Index N	data	data.

Funcs NextSteps ⇒ Proyecto de DATA Personal p/ seguir Aprendiendo

↳ Kaggle
↳ Finance.

IBM - DATA SCIENCE — COURSERA.

- PROF. CERTIFICATE.

CURSO 1 = WHAT IS DATA SCIENCE?

STARG VIDEK

CURSO 1 - SEMANA 1

5 V'S OF BIG DATA

1) VELOCITY
2) VOLUME
3) VARIETY
4) VERACITY
5) VALUE

1) A LA QUE DATA SE ACUMULA. muy RÁPIDO y SIN PARAR.
2) la ESCALA de los datos y el ACUERDO en la CANTIDAD de DATOS ALMACENADOS.
3) diversidad. ESTRUCTURADA y NO ESTRUCTURADA de TWEETS, personas, procesos, ETC.
4) QUALITY and ORIGIN.
5) HABILIDAD y NECESIDAD de CONVERTIR data en VALOR. (NO SOLO $).

(#1) 80% de los DATOS son NO ESTRUCTURADOS.

→ HADOOP ¿qué es? → cloud. COMA. lol como WS AZURE, AWS, ETC, para DATA.

STARG VIDEK

CURSO 1 - SEMANA 2

→ AUTOMATICALLY AND PROGRAMATIC.

DATA MINING → ANALYZING & MODELING DATA REVEALING PREVIOUSLY HIDDEN INFO

1) SET UP GOALS FOR THE EXERCISE. IDENTIFY KEY QUESTIONS THAT NEED TO BE ANSWERED. CONSIDER COSTS & BENEFICIOS DEL EJERCICIO. DETERMINE IN ADVANCE THE EXPECTED LEVEL OF ACCURACY AND USEFULNESS OF THE RESULTS. LA DATA ES UNA RESTRICCIÓN.

2) SELECTING DATA A VECES LA DATA NO ESTA DISPONIBLE AL MINADO. EN ESTOS CASOS ES NECESARIO IDENTIFICAR OTRAS FUENTES Y/O CREAR NUEVAS FORMAS DE RECOLECCIÓN EL OUTPUT DE TODO EL PROCESO, SU ACCURACY Y ETC DEPENDE EN GRAN MEDIDA DE LA CALIDAD DE MISMOS DATOS.

3) PREPROCESSING. DATA ELIMINAR DATA Q NO ESTE COMPLETA O Q ESTE MAL CARGADA DEVELOP A FORMAL METHOD OF DEALING W/ MISSING DATA AND DES IF IT IS RANDOMLY OR SYSTEMATICALLY MISSING.

4) Transforming. Data determine the appropriate formats to store data. Reducir el Nº de vars. al mínimo de los cuales se necesita explicar el fenómeno. Principal Components Analysis

5) Storing Data must be stored in a format that gives unrestricted and immediate read/write privileges to the data scientist. Servers y Backup and Store. + Security and Privacy.

6) Data mining data mining algo, ML, data viz., etc.

7) Evaluating mining Results formal evaluation. Testing predictive capabilities, "In Sample forecast" Sharing w/ Stake Holders. ¡Iterate!

STARG VIDEK

→ ML VS DEEP LEARNING

- MACHINE LEARNING = A SUBSET OF AI THAT USES COMPUTER ALGORITHMS TO ANALYZE DATA AND MAKE INTELLIGENT DECISIONS BASED ON WHAT IT HAS LEARNED, WITHOUT BEING EXPLICITLY PROGRAMMED. TRAINED W/ LARGE SETS OF DATA, LEARN FROM EXAMPLES, NOT FOLLOW RULES-BASED ALGORITHMS.

- DEEP LEARNING: A SPECIALIZED SUBSET OF MACHINE LEARNING THAT USES LAYERED NEURAL NETWORKS TO SIMULATE HUMAN DECISION MAKING. CAN LABEL AND CATEGORIZE INFO AND IDENTIFY PATTERNS. CONTINUOSLY LEARN ON THE JOB.

- ARTIFICIAL NEURAL NETWORKS TAKE INSPIRATION FROM BIOLOGICAL NN. COLLECTION OF SMALL COMPUTING UNITS CALLED NEURONS, THEY TAKE DATA AND DECIDE. USUALMENTE CONSIDERADAS DEEP LEARNING.

STARG VIDEK

→ AI VS DATA Science

DS USA AI PERO ES EN ESENCIA UN TERMINO Q 'INCLUYE TODO EL PROCESO DE PROCESAMIENTO' DE DATOS.
MIENTRAS Q AI INCLUYE LOS MAEC Q LAS PCS PUEDEN APRENDER Y TOMAR DECISIONES INTELIGENTES.

⇒ **REGRESION** (NO VIMOS NADA ACA....)

⇒ **CURSO 1 - SEMANA 3**

- HOW IS DS SAVING LIVES? ML Y DL PARA PREDICCIONES ECONÓMICAS EN ARGENTINA??

"# IF YOU'RE UNABLE TO MEASURE SOMETHING, YOU ARE UNABLE TO IMPROVE IT."

⇒ Companies Should be capturing DATA!!

"DS will change the way Companies compete and operate"

STARG VIDEK

→ THE FINAL DELIVERABLE

~ 1000 - 7000 words. * 4 communicating findings to stakeholders. * vs academia. in business world, the FD ~ 1500 words w/ many viz or maybe several hundred pages as well.
much more powerfull, because of its interactive, than a Power Point presentation.

Before Analysis started their report and investigation, they should (must) have discussed the scope of the FD. Not doing so would likely to result in a poor quality FD. (decide scope, then look for data and Analytics).

→ Careers and Recruiting for DS

. How to become a DS? High end DS are mostly PhD's. Come out of Physics, Stats, have to have CS Background, math bg, DB's, Probability.

• for a math ⇒ • know how to program
• Algebra, Calculus • Prob, Stats. • DB's.

sorg-industrial ⇒

#THINKERER. Kickoff.
| BUILDER.

STARG VIDEK

- As you go up in the field, you really need to know A LOT about CS theory, STATS, PROB.

- One of the ways you learn things is: you do them! ;).

⇒ RECRUITING FOR DS: given the pool of Applicants you have, who has the most RESILIENCE with your firm's DNA? Who's passionate?* you can teach Analytics skills. Curiosity. do they have a sense of humor? lol. At last, Technical skills. * STORY TELLING.

Someone who is Relatable. he/she will need to Relate with many departments.

⇒ The Report Structure, it depends on the length of the document. this varies depending on its purpose.

- COVER PAGE
- TABLE OF CONTENTS
- METHODOLOGY
- CONCLUSION
- ABSTRACT
- INTRODUCTORY
- RESULTS
- LITERATURE REVIEW
- DISCUSSION

STARG VIDEK

Preg's. p/ exclusive w/ final report

1 - Have you told readers at the outset, what they might get by reading your paper?

2 - Have you made the aim of your work clear?

3 - Have you explained the significance of your comission?

4 - Have you set your work in the appropriate context by giving sufficient background (refs) to your work?

5 - Practicality? usefulness of your work?

6 - Future development that might result from your work?

7 - Have you structured your paper in a clear and logical fashion?

Fin curso 1
± 94 %

CURSO 2: TOOLS FOR DATA SCIENCE

STARG VIDEK

- Python
- R
- SQL
- OTROS.

DATA MANAGEMENT — PROCESS OF PERSISTING AND RETRIEVING DATA. (BBDD)

↓

DATA INTEGRATION AND TRANSFORMATION — EXTRACT TRANSFORM AND LOAD

↓

DATA VISUALIZATION

↓

MODEL BUILDING — ML y DL

↓

Model deployment

↓

Model Monitoring y Quality

- EXECUTION ENVIRONMENTS
- DATA ASSET MANAGEMENT
- CODE "....."
- DEV. ENVIRONMENTS (IDE)
- FULLY INTEGRATED VISUAL TOOLS

→ SHARING ENTERPRISE DATA → DAX = DATA ASSET EXCHANGE, TOOL de IBM.

→ MACHINE LEARNING AND DL MODELS

- Identify Patterns.
- Must Be trained
- Can then Predict, decide, ETC.
- 3 main types
  - ↳ supervised learning
  - ↳ unsupervised learning.
  - ↳ Reinforced learning.

• Supervisado = se Human provee los resultados correctos, Labeled data y correct output

SE USA PARA → Regresión
→ clasificación.

- Regression → - Predict real num. vals

- Classif → Classify things into categories

- **No Supervisado**

La data no es usted. El modelo tiene q aprender a identificar patrones solo con la info de la data en sí.

↳ Clustering (dividir en sub grupos parecidos)

↳ Anomaly Detection (anomalías)

- **Reinforced**

Parecido a como aprenden los seres vivos. prueba y error, premios. Chess go, etc.

STARG VIDEK

DEEP LEARNING

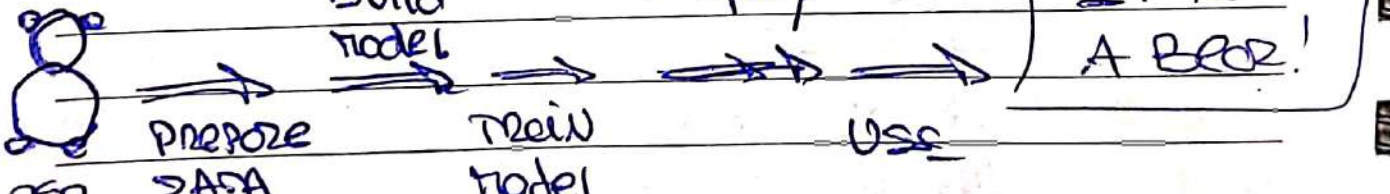
→ Intenta reproducir la forma en la q los humanos aprendemos.

↳ NLP, Image processing, Audio y Video Analytics, time series forecasting,

↳ Suele requerir large DATA SETS of labeled DATA p/ training. ES intensa computacionalmente.

↳ Build from scratch or public repo.

- TENSORFLOW - KERAS.
- pyTORCH



ITERATIVO.

LONG TIME TO VALUE

↳ Contiene muchos veces USAR PRE-TRAINED models →

→ Poder usar AWS (Model Asset xclarge) de IBM.

- SEMANA 2 : Jupiter Notebooks

Allows to combine text, code, tables, graphs, output y more.

1. Jupyter Lab → es como el R studio de Jupyter (:).

~~Aprendizaje~~

- EDA = Exploratory Data Analysis. Graphs, understanding the data.

(#) Se vieron cosas de Jupyter y R studio. No le di mucha bola a R pero ya lo use, y pinta aprenderlo bien en estadísticas.

STARG VIDEK

Git y GitHub

Short Glossary

- SHH Protocol = Method for remote secure login from one pc to another.
- Repository = Folders of the project set up for version control
- Fork = A copy of the Repo.
- Pull Request = The process that you use to request that someone reviews and approves your changes before they become final.

Commands

- Init
- Add
- Status
- Commit
- Reset ⇒ undo changes
- Log ⇒ Browse previous changes.
- Branch ⇒ Create an isolated env. 4 changes
- Checkout ⇒ See y change branches.
- Merge ⇒ put everything back together

Pull Req = Lo hacés cuando hiciste cambios y querés q los demás los unan pq suman al master. Esas propuestas tus cambios.

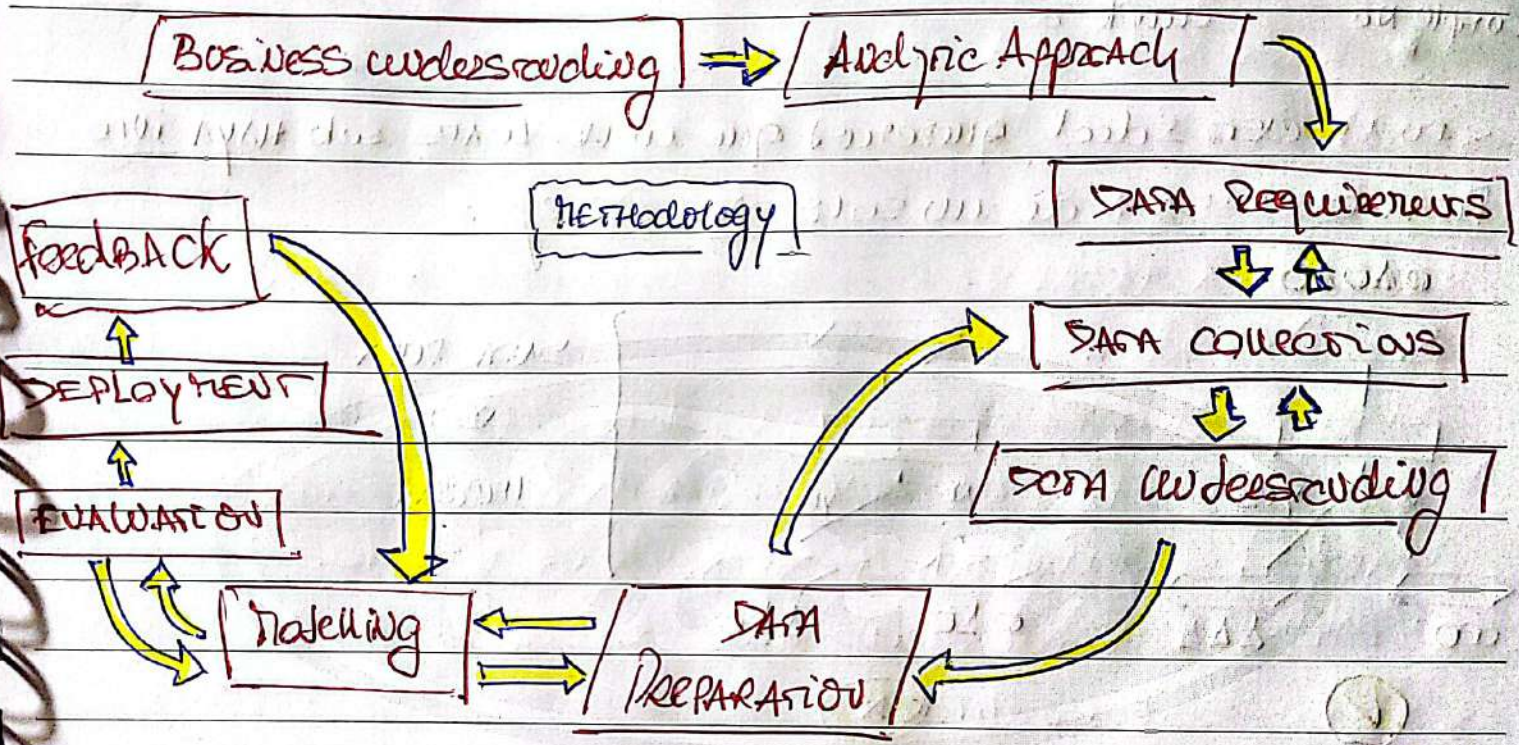
⇒ Semana 3 = IBM Watson Studio

Seguí con los cursos de Data Science en otro cuaderno (Ford Grande).

→ IBM - DATA SCIENCE METHODOLOGY - IBM DS P. CERIF.

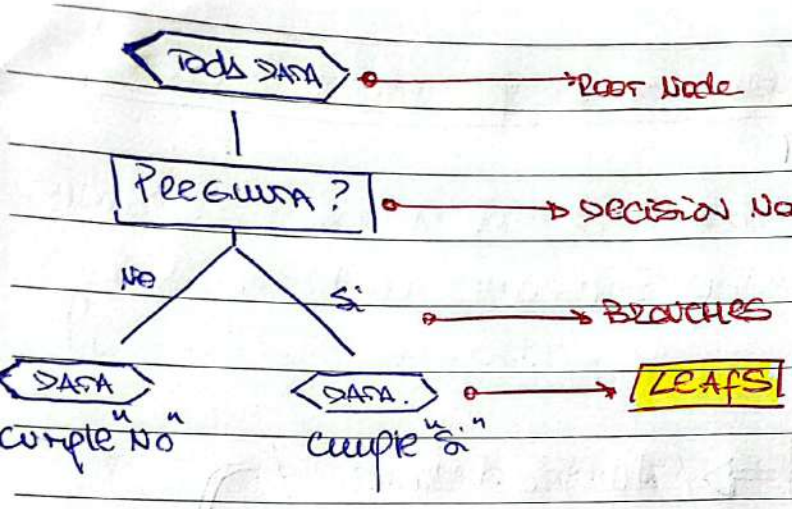
□ FROM PROBLEM TO APPROACH

- CRISP-DM = METODOLOGÍA CREADA PARA APLICAR DS. MUCHO ÉNFASIS EN LA IMPORTANCIA del BUSINESS UNDERSTANDING Y EN HACER PREGUNTAS.

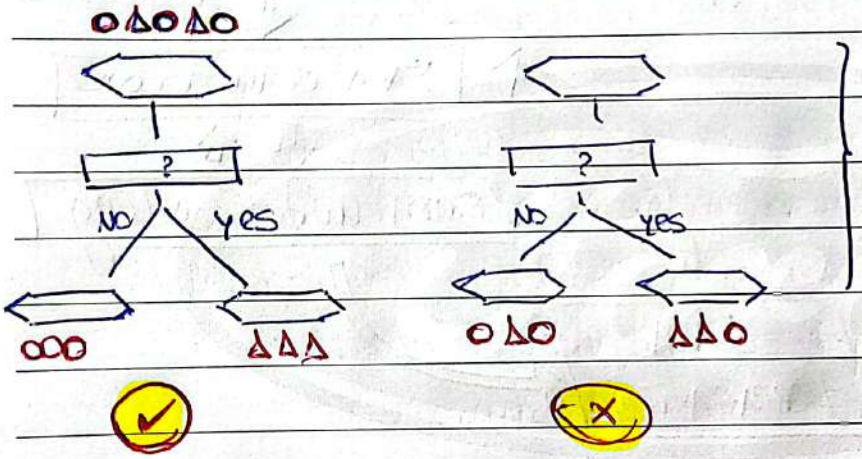


→ DECISION TREES

- Built using RECURSIVE PARTITIONING (MUCHOS SUBPROBS) TO CLASSIFY DATA
- WHEN PARTITIONING, DECISION TREES USE THE MOST PREDICTIVE ~~EFFICACE~~ FEATURE TO SPLIT THE DATA (DE MENOR A MAYOR EN DESCENDANTE?)
- PREDICTIVENESS is BASED ON DECREASE IN ENTROPY - GAIN IN INFORMATION, OR IMPURITY.



→ En el caso ideal, queremos que en las leafs solo haya info pura. Osea, data de un solo tipo. Por ej:



A tree stops growing at a node when

- Pure or nearly pure
- No remaining VARS. to subset the data.
- It has grow passed a preselected size limit

From Requirements to Collection

→ Al igual q en la cocina, los ingredientes (data, tipos) serán clave para un Buen Resultado

Es importante saber qué datos vamos a necesitar para resolver, y cómo hacernos con esos datos.

From Understanding to Preparation

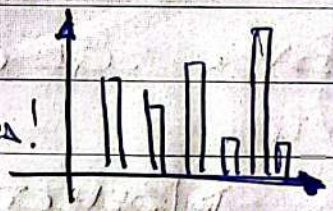
→ Is the data that you collected representative of the problem you need to solve?

- Descriptive statistics → Univariate statistics, Pairwise ~~stat~~ correlations, Histograms

$$[\dots] \quad b(a) + \sum_{k=1}^{N} \frac{1}{k!} \frac{d^k}{dt^k} \Big|_{t=0} \quad f(u(t)) + \int_0^1 \frac{(1-t)^N}{N!} \frac{d^{N+1}}{dt^{N+1}} f(u(t))\, dt$$

$f_{xy}(x,y) \;/\; f_{xy}(x,y) = f_x(x)\, f_y(y)$ [...]

¡Probl y Estadística! (y)



- Iterative data collection and understanding [Etc]

From Modelling to Evaluation

- Plantear un modelo / de Estadística / ML
- Evaluarlo
- Tunearlo. (ie cambiando parámetros pa q mejore en resultados)

FROM DEPLOYMENT TO FEEDBACK

una vez q el modelo está deployado lo mandamos a prod y explicamos a stakeholders + feedback. Después del 1er año se revisa el modelo.

→ DATABASES AND SQL FOR DATA SCIENCE ←

- BASICS. "Structured Query Language"
- RELATIONAL DATABASES

→ WHAT IS A DATABASE ?

- Repository of data
- Adding, modifying and querying data
- different kinds of databases store data in ≠ forms.
- TABLE ⟷ RELATIONAL-DB ⟷ ROWS y COLS.
- DBMS = data base management system. (RDBMS = relational)

→ BASIC SQL commands = CREATE, INSERT, SELECT, UPDATE, delete

→ SELECT STATEMENT ←

- "SELECT * FROM <TABLENAME>" .

¿que pase si quiero filtrar ? → "WHERE"

- SELECT BOOK_id, TITLE FROM BOOKS
  WHERE <predicate> → true/false/unknown
  ↓ ej
  BOOK_id = "B1"

OPERATORS →	OP	SINTAX
	Equal	=
	Greater	>
	lesser	<
	gleq	>=
	lleq	<=
	Noteq	<>

Los statements q son DML son los q permiten modificar la DATA "DATA manipulation Lang."

(#)

DDL "data def. lang" Define, change DBP...

→ Select Statement's Expressions

- COUNT() = Retrieves N° of Rows matching the query criteria.
  - SELECT COUNT(*) FROM TABLE.
  - SELECT COUNT(COUNTRY) FROM MEDALS WHERE COUNTRY = "CANADA"
- DISTINCT() = Remove duplicate values from Query, Retrieve unique values.
  - SELECT DISTINCT COLUMNAME FROM TABLENAME.
- Limit() = N° max de Rows en el Query
  - SELECT * FROM TABLE WHERE year = 2018 LIMIT = 5

→ INSERT STATEMENT ←

Agregar nuevos filas a la DB.

- INSERT INTO <TABLE> <(COLUMNName,...)> Values ([value], ...)

⇒ Luego q poner igual cant values y Cols.

→ Se puede agregar de a ⊕ de una row a la vez!

```
INSERT INTO Actor (Id, Name, Email)
Values
    (A1, Chung, @)
    (Ad, Joe, @2)
    :
```

→ UPDATE and DELETE STATEMENTS ←

→ UPDATE TABLE → Puede ser ⊕ de uno (y)
SET [[Col Name] = [Value]]
Where [condition]

OJO si no pones el Where cambian todas las filas!!! ⚠

→ DELETE FROM TABLE
Where [condition]

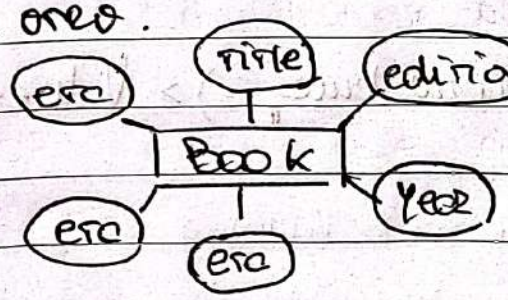
Semana 2

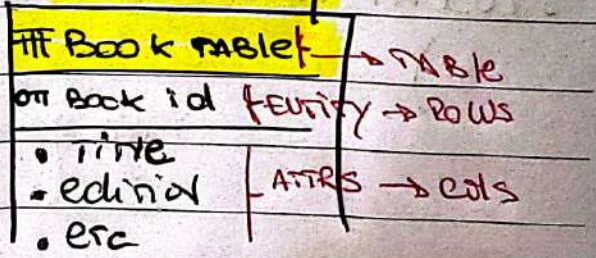
⇒ Relational DB Concepts. Information Model and Data Models.

- Relational model → Allows for data independence.

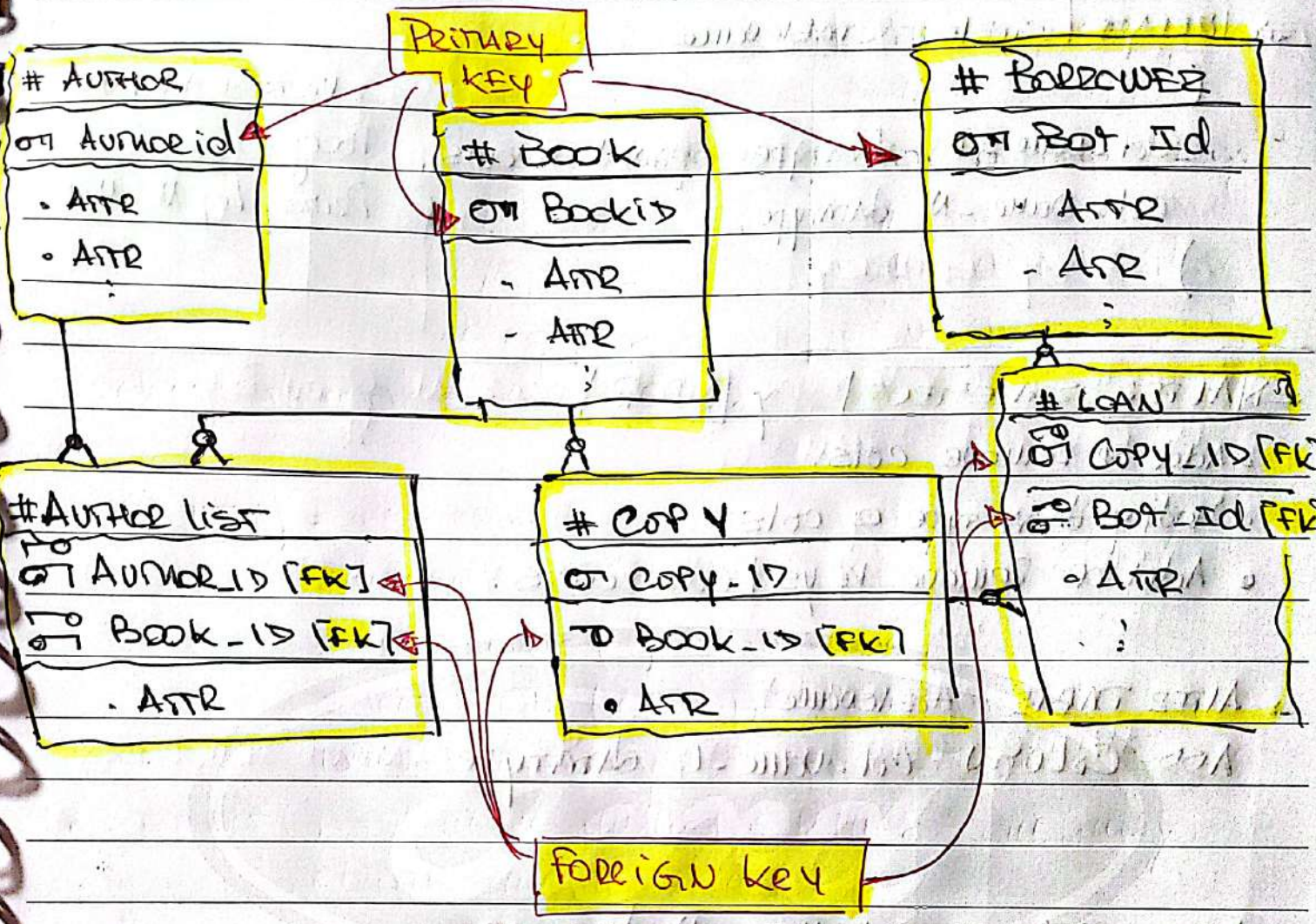
→ Entity Relationship Model = Thinking a DB as a collection of entities. (4 designing DB's). Objects q existen indep. uno del otro.

○ = ATTRIBUTES
▭ = ENTITY





Para Nuestro Ejemplo de la DB de Libros podría verse así:



- PRIMARY KEY = Key una POR TABLA. Asigna valores ÚNICOS a cada fila.
- FOREIGN KEY = Primary keys definidas en OTRAS TABLAS q definen un link (Relación) entre las tablas.

DDL = data def. lang. STATEMENTS (≠ DML)

↳ define, change or drop DATA

↳ ~~CREATE~~, ~~ALTER~~, ~~TRUNCATE~~ ~~DROP~~.

~~ELIMINAR~~ PARTE de TABLA

~~ELIMINAR~~ TABLA.

⇒ CREATE TABLE tablename

(

Col_Name_1 datatype optional_params*,

Col_Name_N datatype.

)

→ Views + Adelants.

Pkeef =

Primary key Not Null.

⇒ ALTER STATEMENT y DROP

- Add or Remove cols.
- Modify data type of cols.
- Add or Remove keys | constraints.

→ ALTER TABLE tablename

ADD COLUMN col_name_1 datatype

⋮

ADD COLUMN Col_Nam_N datatype;

→ ALTER TABLE tablename

ALTER COLUMN col_Name SET DATA TYPE

<datatype>; #000, si ya tiene datos DT, tira error

→ ALTER TABLE tablename

DROP COLUMN telephone_num;

→ DROP TABLE table Name;

→ TRUNCATE TABLE <Name>
IMMEDIATE ; #BORRA TODA LA DATA (FILAS). NO SE puede deshacer

⇒ INTERMEDIATE SQL ⇐

- REFINING RESULTS: String Patterns, Ranges, grouping
- functions, Multiple TABLES, SUB-queries.

↳ USING STRING PATTERNS AND RANGES

¿qué pasa si no se que pone en el where? puede, se q de el nombre empezaba con "R" pero no me acuerdo entero. Como hacemos?

→ WHERE columna LIKE <STRING PATTERN>

% = Representa missing letters. si pongo "R%" estoy diciendo: Los q empiecen con R.

RANGES → SELECT title, pages FROM Book
WHERE pages BETWEEN 290 AND 300

IN → SELECT name, country FROM AUTHOR
WHERE country IN ("AU", "BR") → "sets of values"

↳ SORTING RESULT SETS. ORDER BY. (DEF = ASCENDING)

SELECT title FROM BOOK
ORDER BY title DESC;

TOMB. podes poner "order by 2" donde 2 = segunda col del select statement.

GROUPING Result Sets

```
SELECT COUNTRY, COUNT(COUNTRY)
FROM AUTHOR
GROUP BY COUNTRY;
```

→

COUNTRY	COUNT
ARG	100
BRA	0

TAMB se le pueden Agregar condiciones al GroupBy con la keyword "HAVING"

```
SELECT COUNTRY, COUNT(COUNTRY) AS COUNT
FROM AUTHOR
GROUP BY COUNTRY
HAVING COUNT(COUNTRY) > 4;
```

BASICAMENTE, TE LO FILTRA

→ Built in functions

Si bien podés sacar la data y trabajarla, muchas DBS ya vienen con fcn. y es mucho más rápido usar éstas.
También es posible definir nuestras propias funciones.
Entre ellas hay Sqrt, Length, Ucase, Datetime etc.

→ Subqueries con Nested Selects

```
SELECT COLUMN1 FROM TABLE
    WHERE COLUMN2 = (SELECT MAX(COLUMN2) FROM TABLE)
```

(SELECT MAX(COLUMN2) FROM TABLE) → SUBQUERY

por ejemplo si quiero obtener una lista de empleados que gane más que el salario promedio

```
SELECT * FROM EMPLOYEES
    WHERE SALARY > AVG(SALARY)
```

→ ERROR!!

→ En cambio →

(no todas las Agg Functions pueden correr en el where, por eso usamos N. Select)

```
select * from Employees
    where salary <
        (select Avg(SALARY) from Employees);
```

→ Los N. Select también pueden ir en otros lugares. por ejemplo en Columns. A eso se le llama

↓

- **Column Expression**

```
select emp_id, salary,
    (select AVG(salary) from employees)
        AS Avg_salary
    from employees;
```

- **Derived Tables or Table Expr.** (N. Select en from)

Por ej. si queremos una tabla sin data sensible:

```
select * from
    (select Emp_ID, F_Name - L_Name from Employees)
        AS Emp4ALL;
```

Esto es un ejemplo trivial xq podríamos hacerlo pedido de uno, pero dice que es para ver.

STARG VIDEK

→ Ahora x ~ 3:19 min.
→ Muestra std inicial ≈ 4-5k.
→ 3,7k luego de cumplejo. ↳ total res?

→ Working w/ more than one table

↳ • Sub-queries
• Implicit join
• join operators (inner, outer, etc)

```
SELECT * FROM Employees
    Where DEP_Id in
    (select DEP_ID_DEP from departments
        Where Loc_Id = "L002");
```

→ Un join implicito o full join es el que toma todo de las 2 tablas (0 ms)

```
select * from employees, departments;
                 TABLA 1      TABLA 2
```

→ Python + DB con SQL

DBAPI: → poder usar Python pi interactuar con cualquier DB con el mismo código.

Conceptos clave → • DATABASE CONNECTIONS

• CONNECTIONS OBJECTS

→ Connection Objects

- database connections
- manage transactions.

→ Cursor Objects

- Database queries
- Scroll through result set.
- Retrieve results.

from database import connect.

```
CONN = CONNECT('db', 'user', 'pwd')
CUR = CONN.CURSOR()

Cur.EXECUTE('select * from mytable')
RESULTS = CUR.FETCHALL()

Curs.close()
CONN.close()
```

STARG VIDEK

→ Creating tables, loading data and querying

SQL magic (Jupiter)

↳ Nos deja correr SQL "directo" de Python de una con 2 (%%) toma toda la celda.

%.sql SELECT * FROM TABLENAME

Drawback = No puedo elegir cuando cerrar la conexión pq libero recursos.

"Loaders module"

→ **Views** forma alternativa de mostrar la data de una tabla

- Pueden incluir columnas de multiples tablas o views.
- Una vez creada se puede hacer queries como una tabla
- Solo la definición de la view se guarda no la data.

STARG VIDEK

Views con

- Show part of the data (sensitive, etc)
- Combine two or more tables in meaning full ways.
- Simplify Access to data (das permisos a la view y no a las tablas)

→ nombres x ri

```
CREATE VIEW <Name> (<colAlias 1>,
  <colAlias 2>, ... <colAlias N>)
    AS SELECT <col1>, <col2>, .. <col N>
    FROM <TABLE NAME>
    WHER <predicate>;
```

→ **SQREDI PROCEDURES** Ser de SQL's guardados en el servidor. En vez de mandar todo el tiempo de a uno, mando uno solo que corre todos los guardados pasando el nombre.

Se pueden codear en muchos lang.

Ventajas
- Reuse of code entre Apps
- ⊕ Seguridad
- ⊖ Traffic
- ⊕ performance (data | logic en juntos)

Parametros.

```
CREATE Procedure update_sal (IN EmpNum CHAR(6),
    W Rating SmallINT)
    Language SQL
    BEGIN
      IF Rating = 1 THEN
          UPDATE employees
            SET Salary = Salary * 1,10
            WHERE emp-Id = emp-NUM;
      ELSE
          Update employee
            SET salary = Salary * 1,05
            WHERE emp-id = emp-Num;
      End if;
    END
```

↳ CALL UPDATE_SAL ('E1001', 1)

⇒ **ACID TRANSACTIONS** UNIDAD INVISIBLE DE TRABAJO. CONSISTE DE 1 O MÁS SQLS. SUCCESS ↔ SUCEDEN TODAS O FALLAN O NINGUNA

Acid

→ ATOMIC = All changes must be performed successfully or not at all

→ CONSISTENT = Data must be in a consistent state before and after the transaction

→ ISOLATED = No other process can change the data while the transaction is running.

→ DURABLE = The changes made by the transaction must persist.

→ PARA CORRER UNA Acid

BEGIN → ES IMPLÍCITO, NO HACE FALTA PONERLO EXPLÍCITO

UPDATE .... } 1

UPDATE ... } 2

ROLLBACK. } N