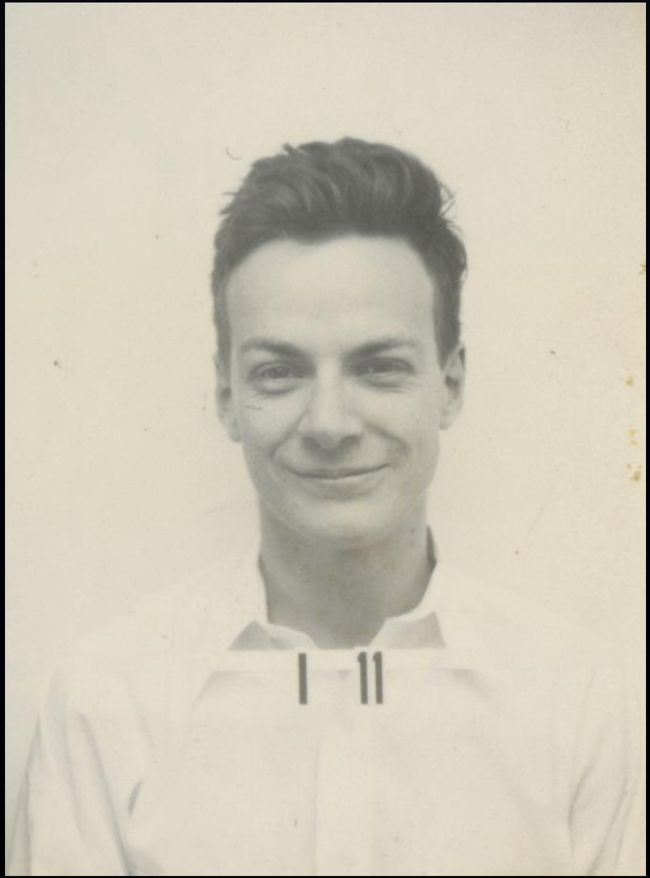
data engines

must grok ‘em all

teach / learn

this dude



teach / learn

it is hubris to think that anyone can come here and **teach** you

-Richard Feynman*

** (I think, I don't know who, maybe someone else said it…)*

teach / learn

no one can teach you

but you **can** learn

teach / learn

the best anyone can hope for is to inspire you to learn that's the only grace I aspire to today

one

PERCEPTION

PERCEPTION

PERCEPTION

PERCEPTION

PERCEPTION

PERCEPTION

become aware of something through the senses

intuitive understanding,

insight

PERCEPTION

PERCEPTION

PERCEPTION

PERCEPTION

PERCEPTION

PERCEPTION

become aware of something through the senses

intuitive understanding,

insight

TASTE

AESTHETIC

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION

RICHARD WAGNER

1813 – 1883, GERMAN

RICHARD WAGNER

COMPOSER

CONDUCTOR

POLEMICIST*

THEATRE DIRECTOR

1813 – 1883, GERMAN

RICHARD WAGNER

operas, musical dramas, prose and letters

57 books of music

13 books of prose and stuff

12000 letters and counting!

„the tristan chord“, „the sound wall“, „father of heavy metal“

1813 – 1883, GERMAN

RICHARD WAGNER

writes

'The Artwork of the Future‘ in 1849

(Das Kunstwerk der Zukunft)

1813 – 1883, GERMAN

RICHARD WAGNER

writes

'The Artwork of the Future‘ in 1849

(Das Kunstwerk der Zukunft)

music + drama = art v2.0

1813 – 1883, GERMAN

RICHARD WAGNER

writes

'The Artwork of the Future‘ in 1849

(Das Kunstwerk der Zukunft)

VAUDEVILLE

1813 – 1883, GERMAN

RICHARD WAGNER

writes

'The Artwork of the Future' in 1849

(Das Kunstwerk der Zukunft)

VAUDEVILLE

PARSI THEATRE

1813 – 1883, GERMAN

RICHARD WAGNER

writes

'The Artwork of the Future‘ in 1849

(Das Kunstwerk der Zukunft)

VAUDEVILLE

PARSI THEATRE

BOLLYWOOD

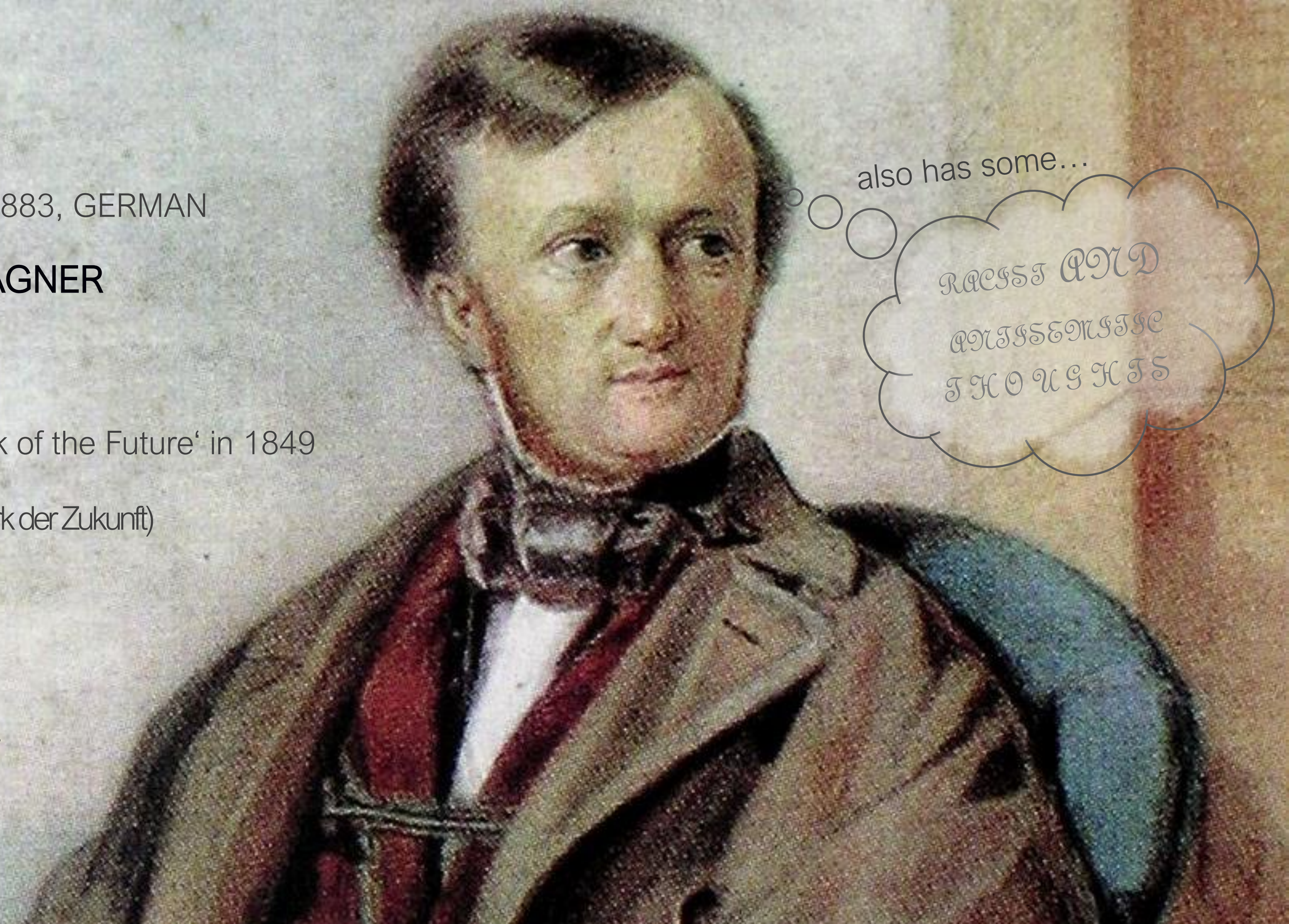
1813 – 1883, GERMAN

RICHARD WAGNER

writes

'The Artwork of the Future‘ in 1849

(Das Kunstwerk der Zukunft)



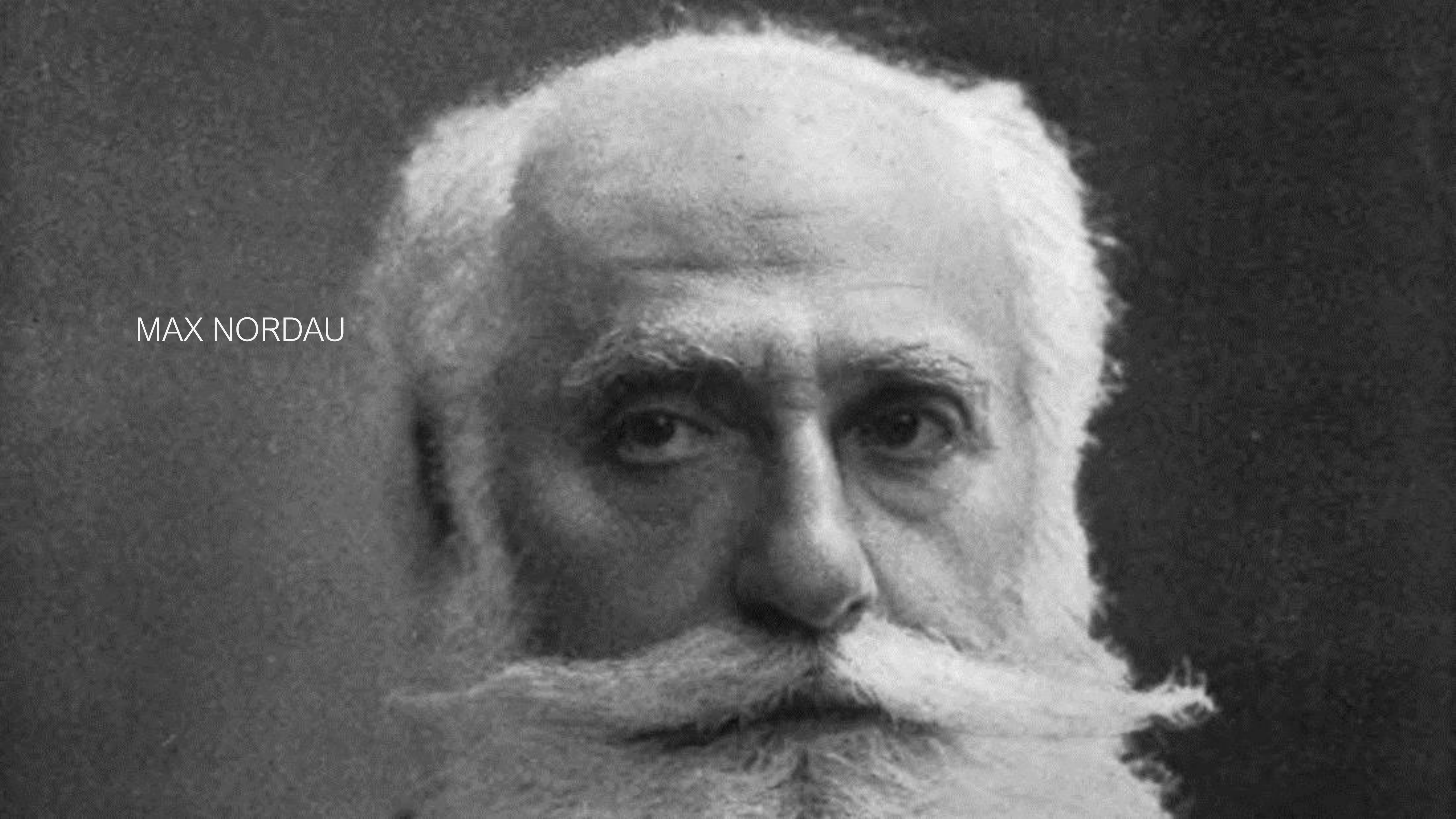
also has some…

RACIST AND ANTISEMITIC THOUGHTS

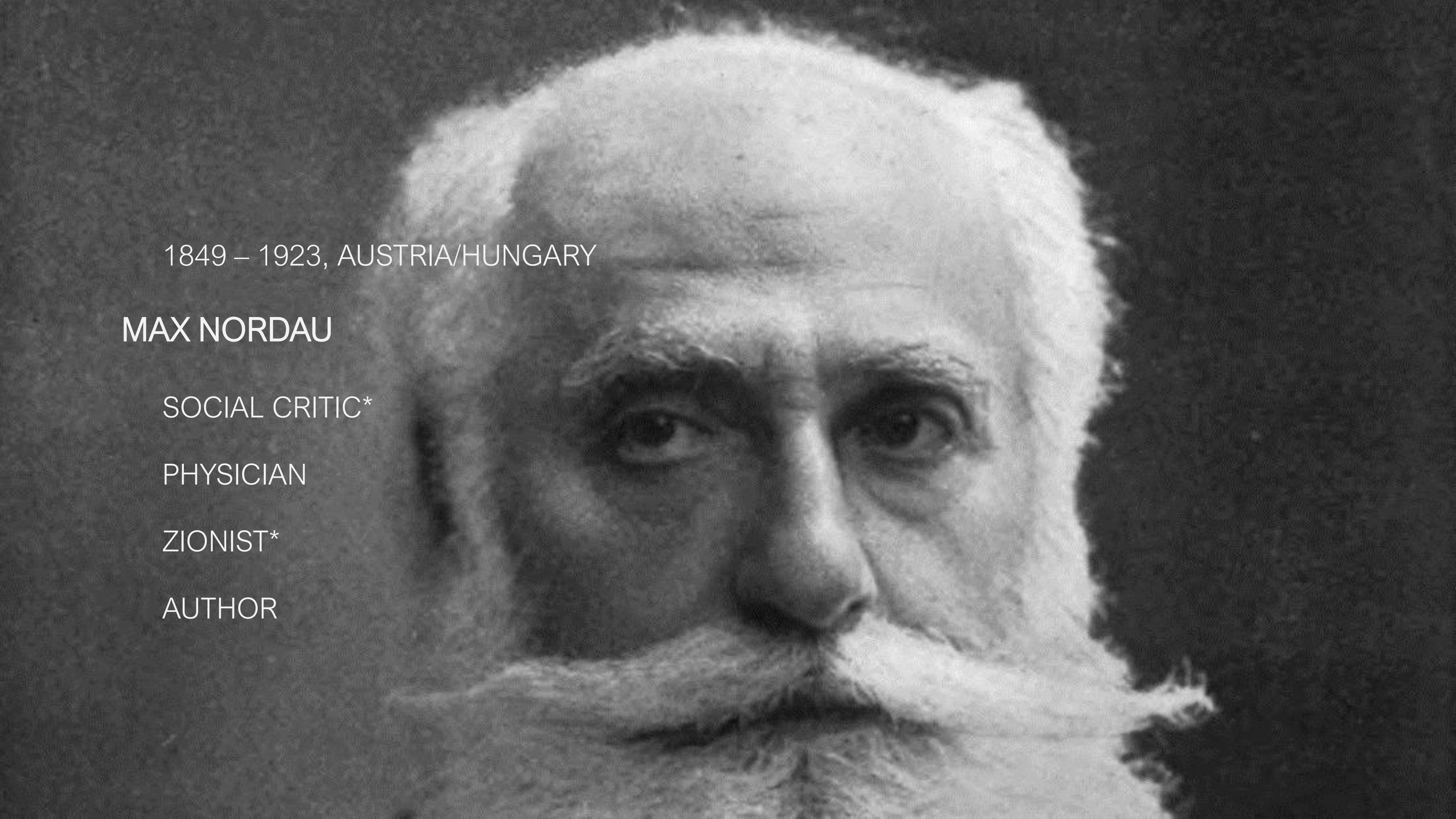
VAUDEVILLE

PARSI THEATRE

BOLLYWOOD



MAX NORDAU



1849 – 1923, AUSTRIA/HUNGARY

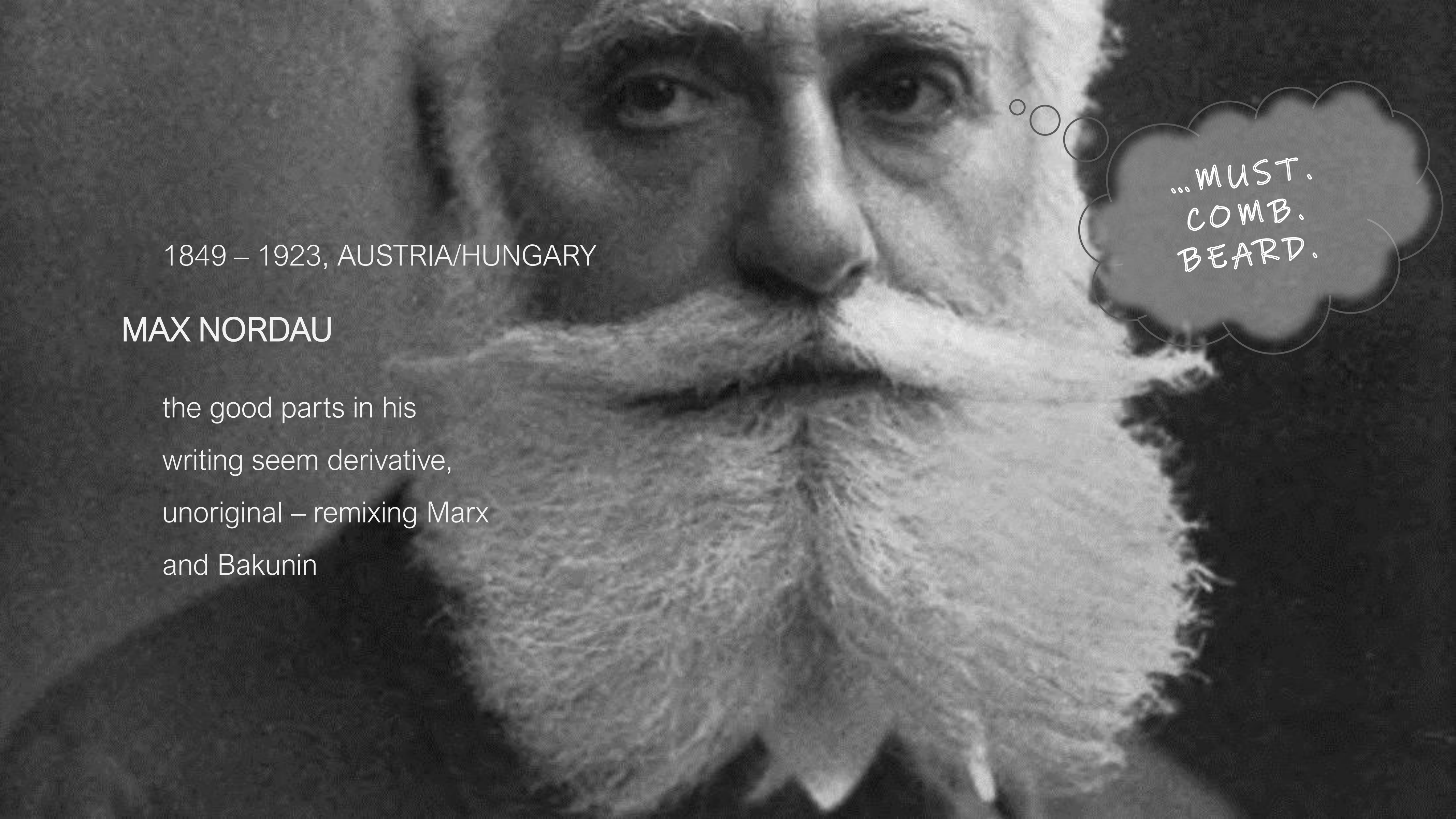
MAX NORDAU

SOCIAL CRITIC*

PHYSICIAN

ZIONIST*

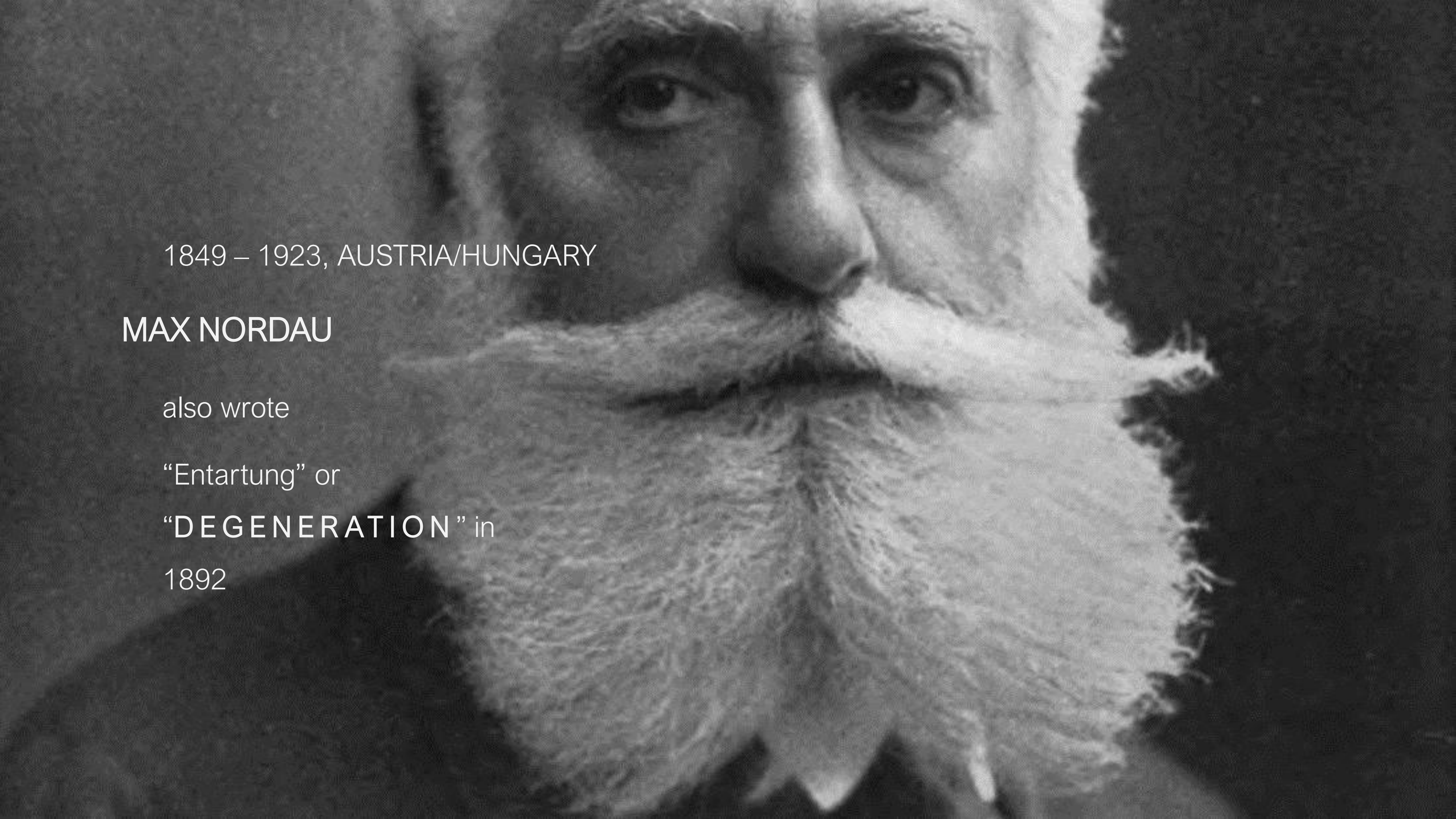
AUTHOR



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

the good parts in his writing seem derivative, unoriginal – remixing Marx and Bakunin



1849 – 1923, AUSTRIA/HUNGARY

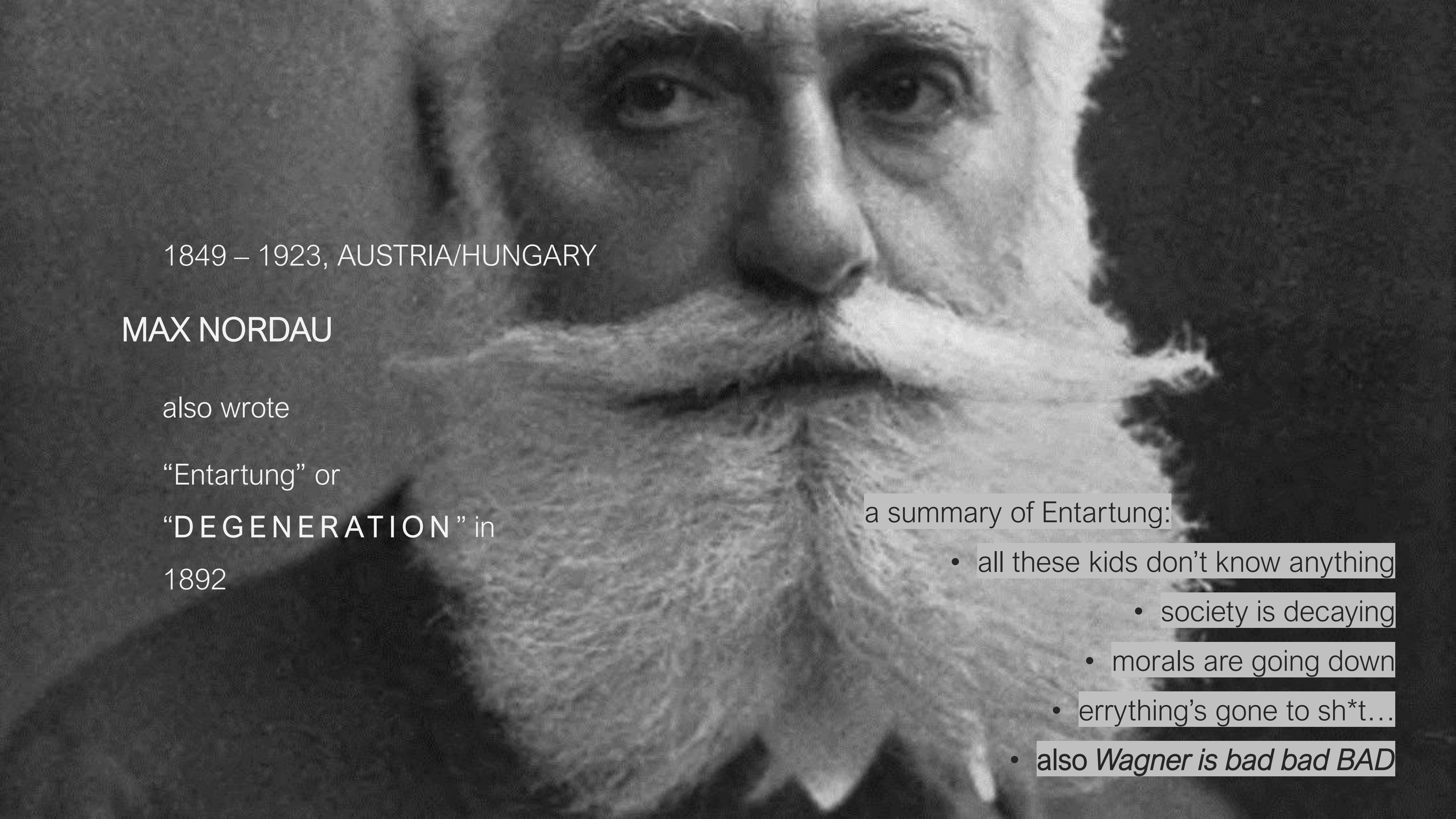
MAX NORDAU

also wrote

“Entartung” or

“DEGENERATION” in

1892



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

also wrote

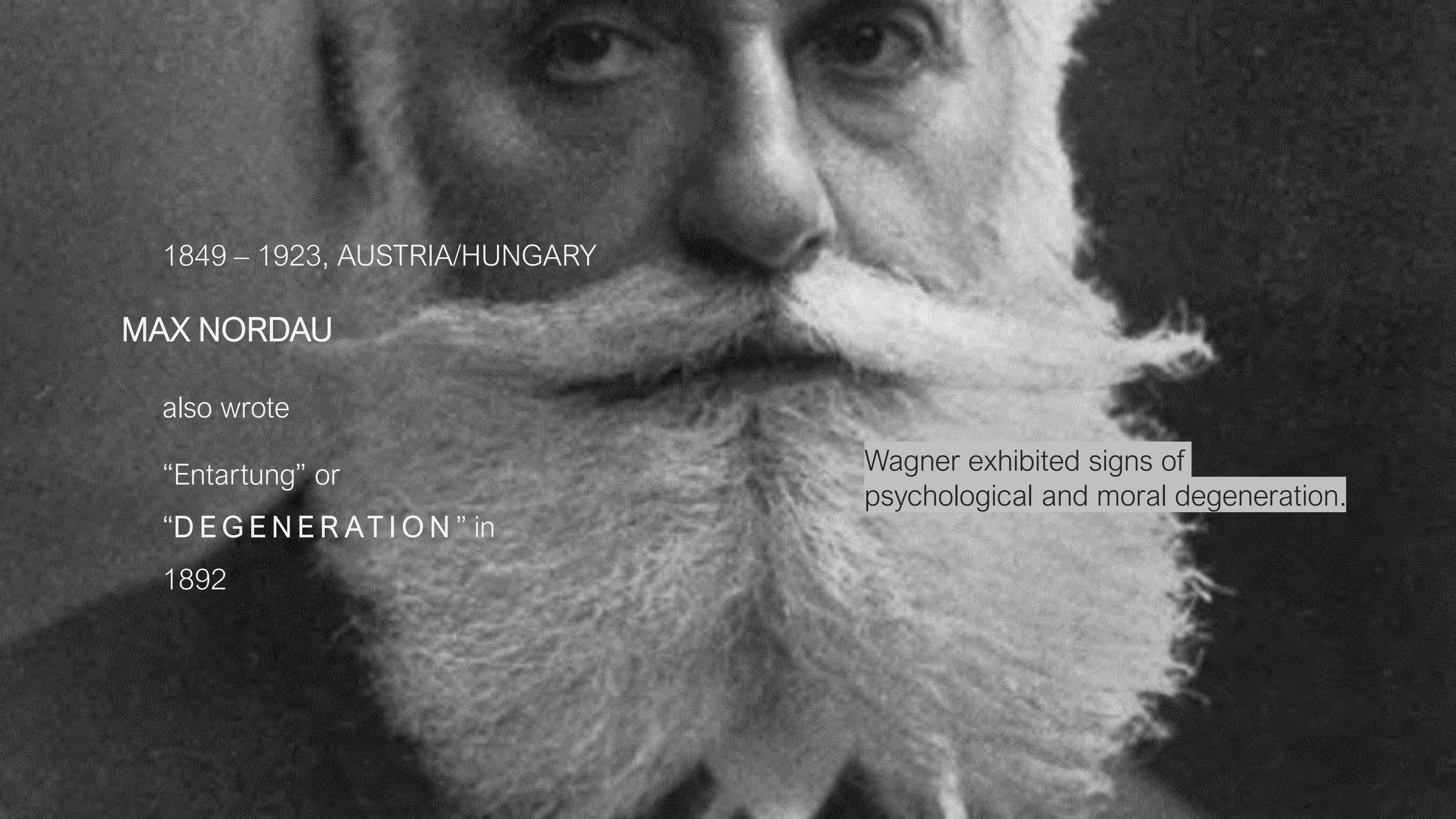
“Entartung” or

“DEGENERATION” in

1892

a summary of Entartung:

- all these kids don’t know anything
- society is decaying
- morals are going down
- errything’s gone to sh*t…
- also *Wagner is bad bad BAD*



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

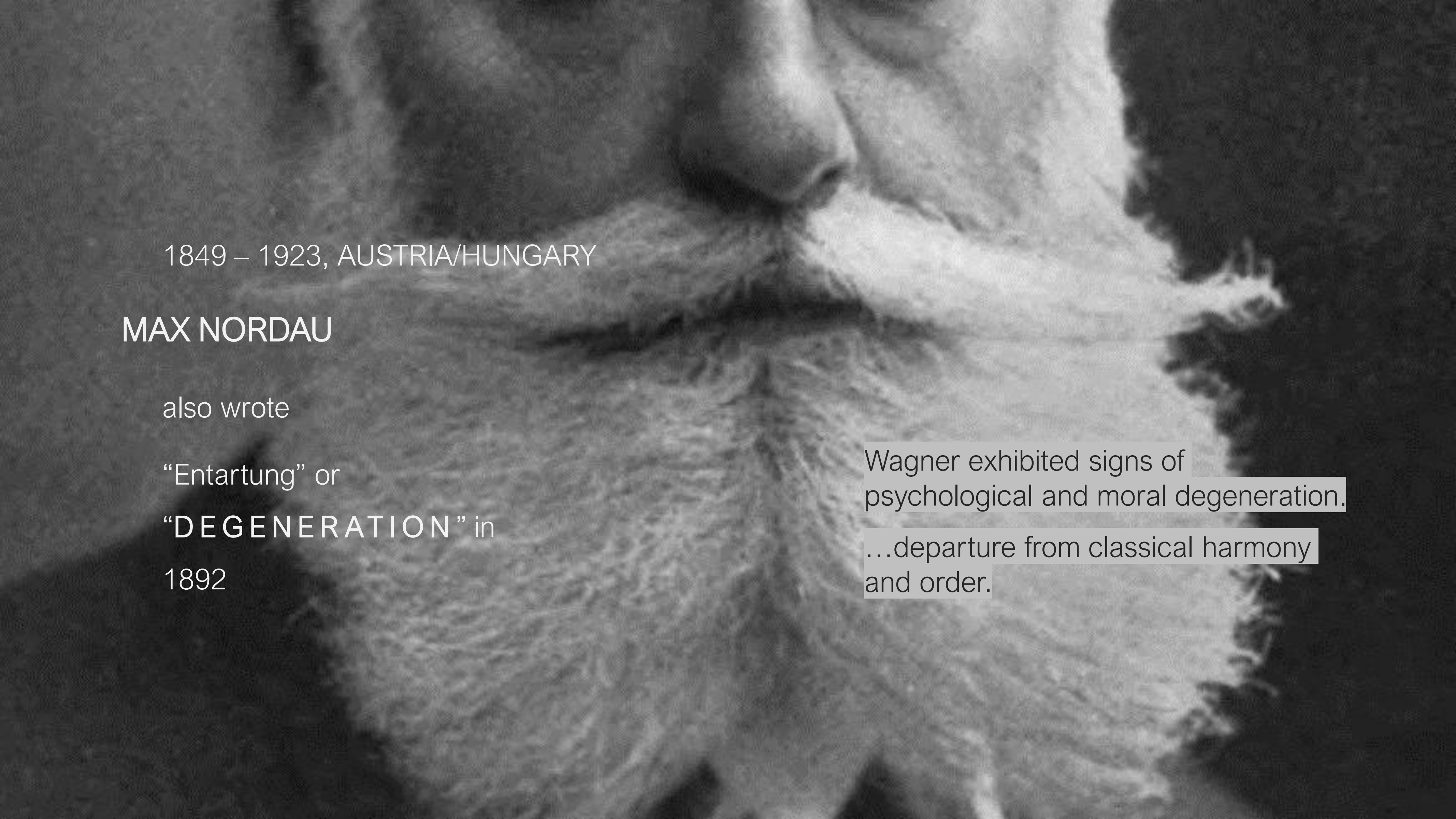
also wrote

“Entartung” or

“DEGENERATION” in

1892

Wagner exhibited signs of
psychological and moral degeneration.



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

also wrote

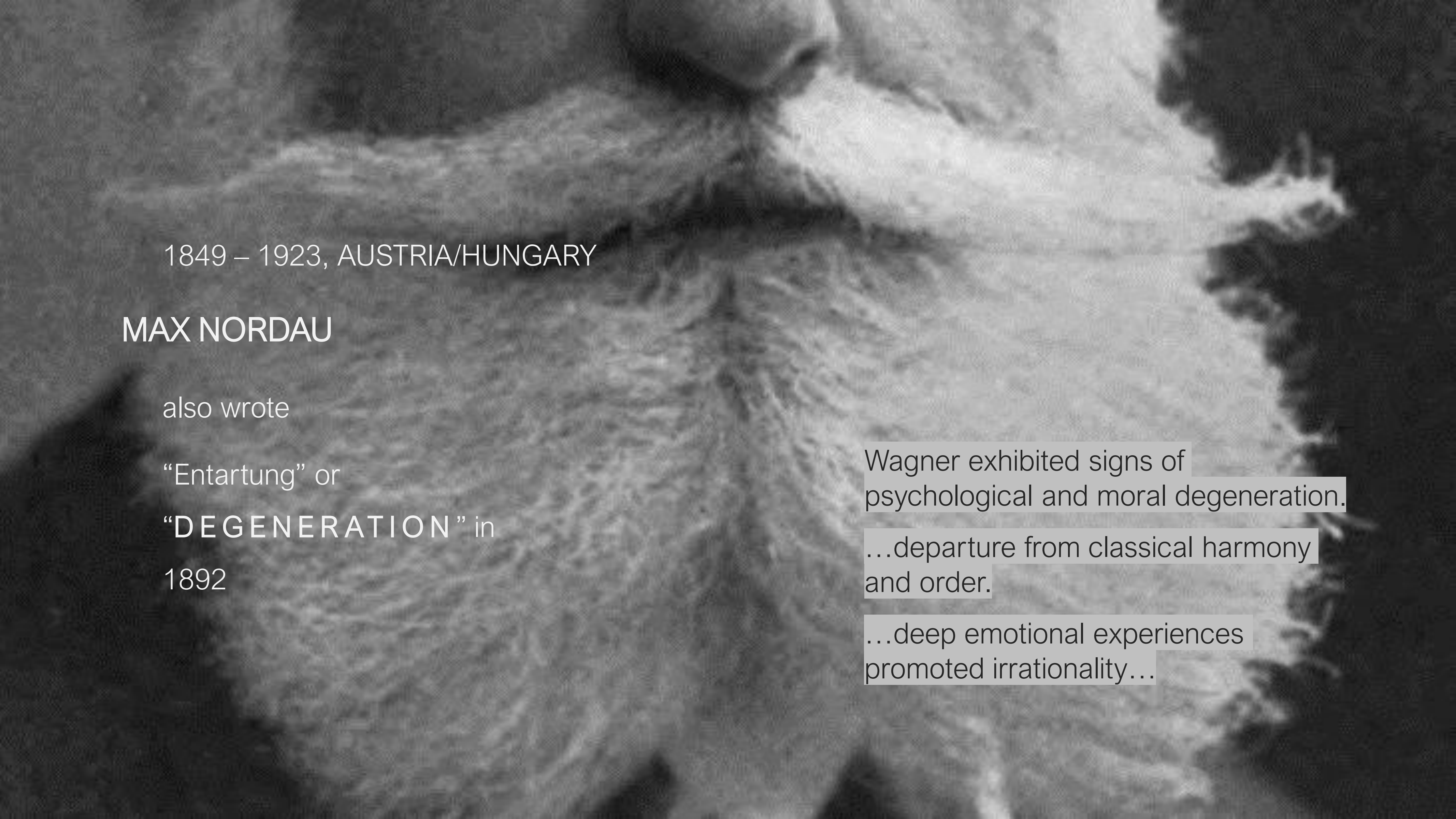
“Entartung” or

“DEGENERATION” in

1892

Wagner exhibited signs of psychological and moral degeneration.

…departure from classical harmony and order.



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

also wrote

“Entartung” or

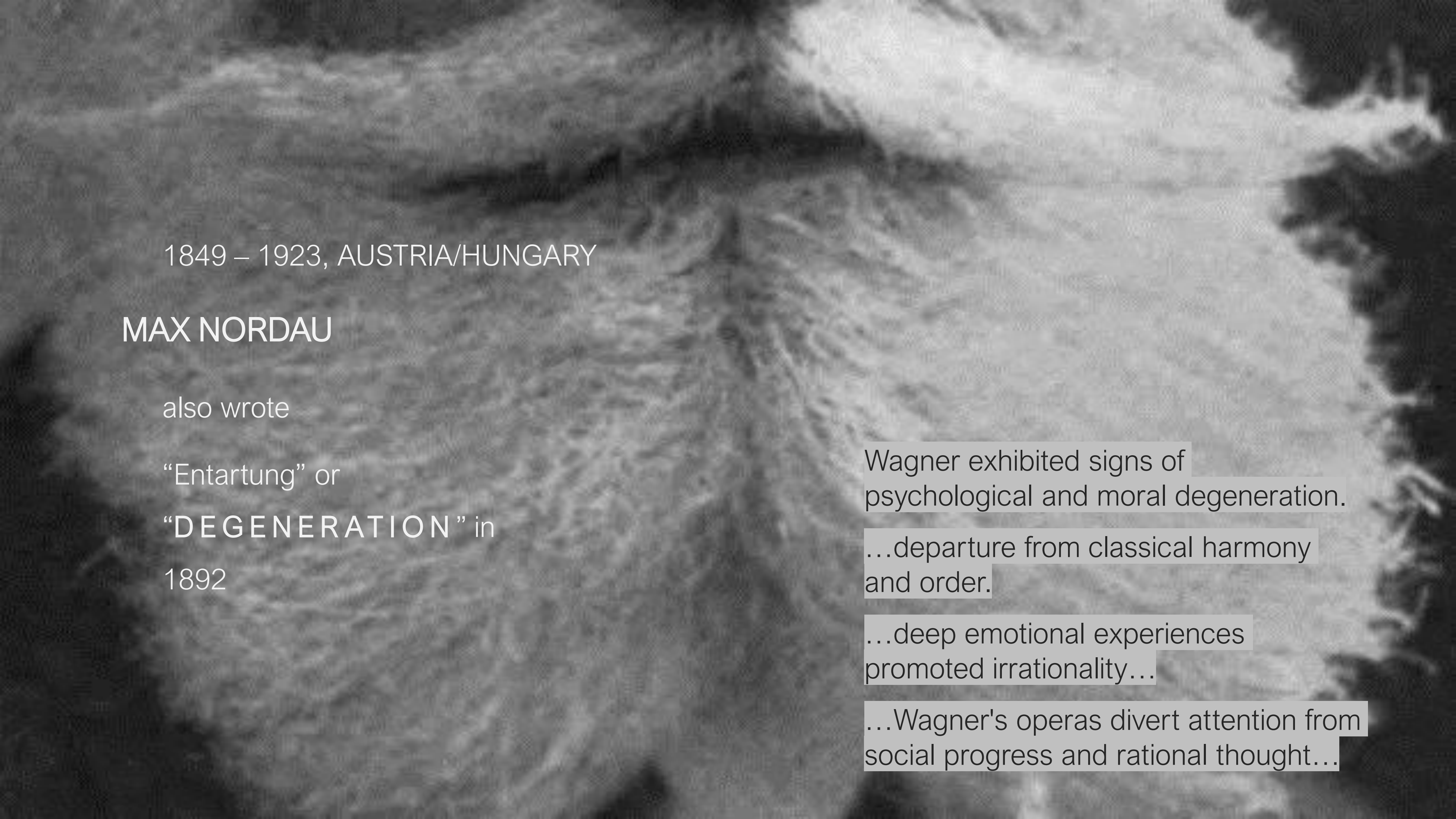
“DEGENERATION” in

1892

Wagner exhibited signs of psychological and moral degeneration.

…departure from classical harmony and order.

…deep emotional experiences promoted irrationality…



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

also wrote

“Entartung” or

“DEGENERATION” in

1892

Wagner exhibited signs of psychological and moral degeneration.

…departure from classical harmony and order.

…deep emotional experiences promoted irrationality…

…Wagner's operas divert attention from social progress and rational thought…

1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

also wrote

“Entartung” or

“DEGENERATION” in

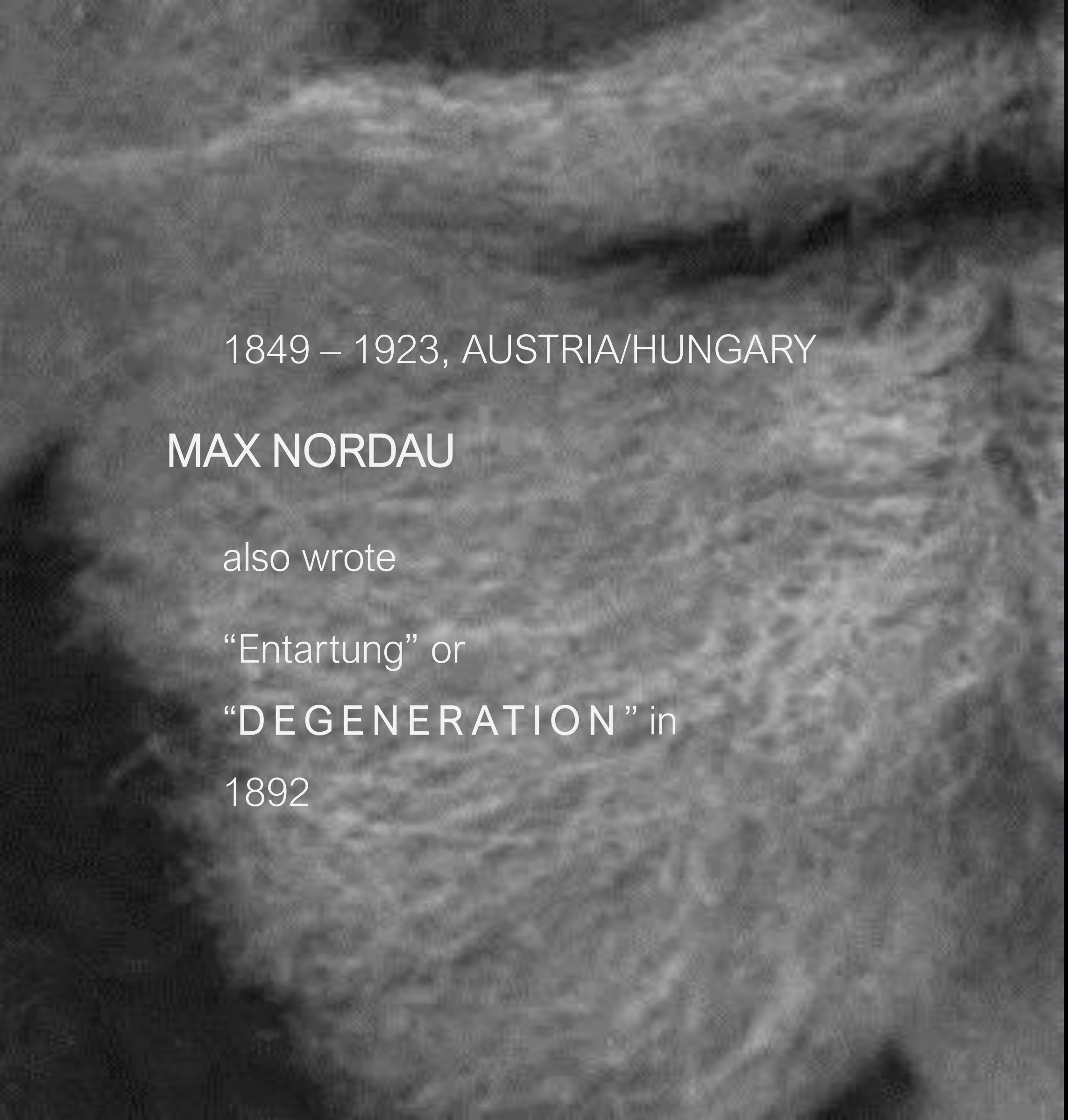
1892

This was obviously DUMB™

If the world worked like that, we wouldn’t get movies, songs, literature, computers, cars, medicine...

*How could Nordau be so misguided?**

* stop thinking about the beard, answer the question!!!



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

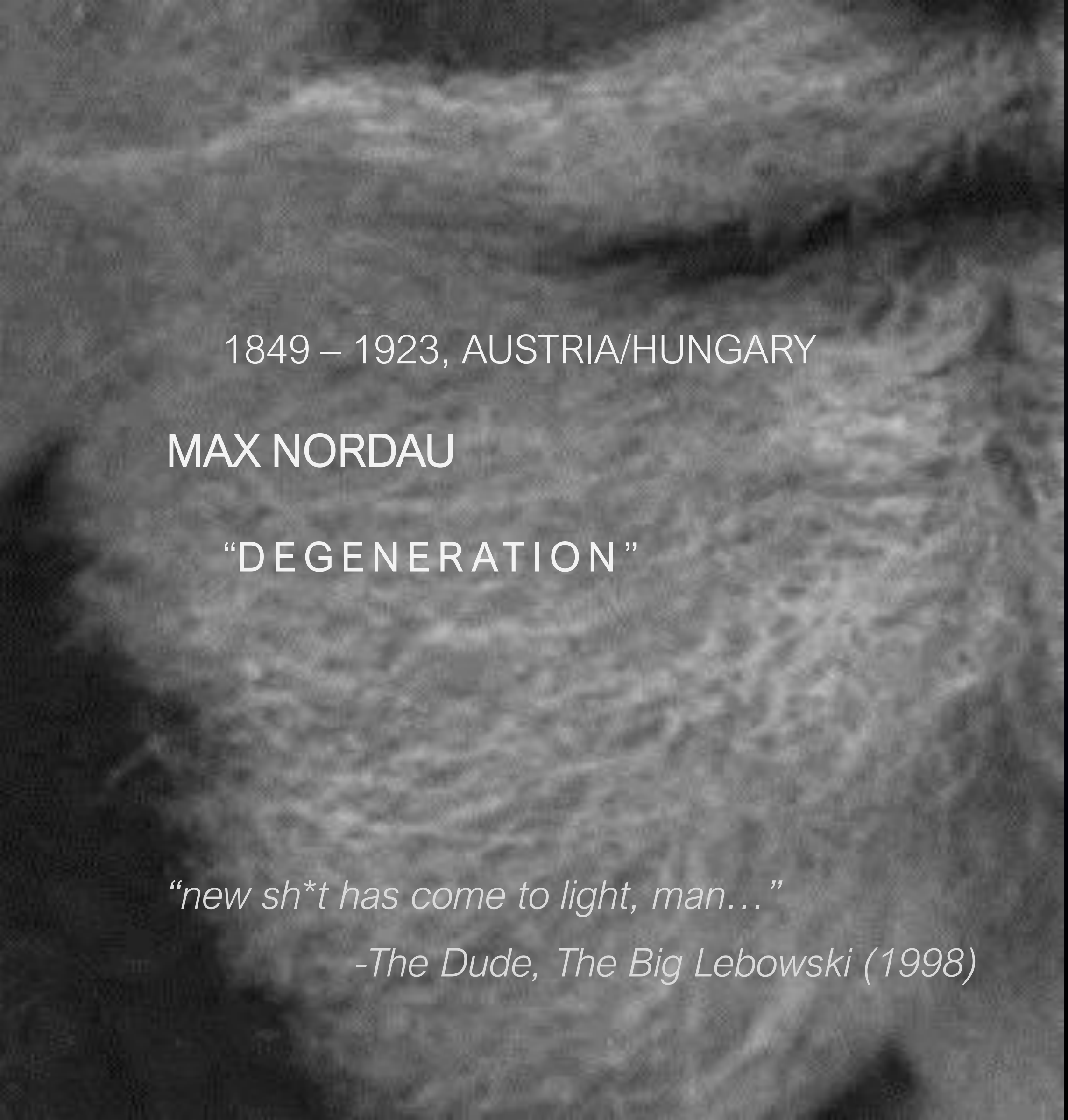
also wrote

“Entartung” or

“DEGENERATION” in

1892

Maybe he saw only half the picture?



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

“DEGENERATION”

*“new sh*t has come to light, man…”*

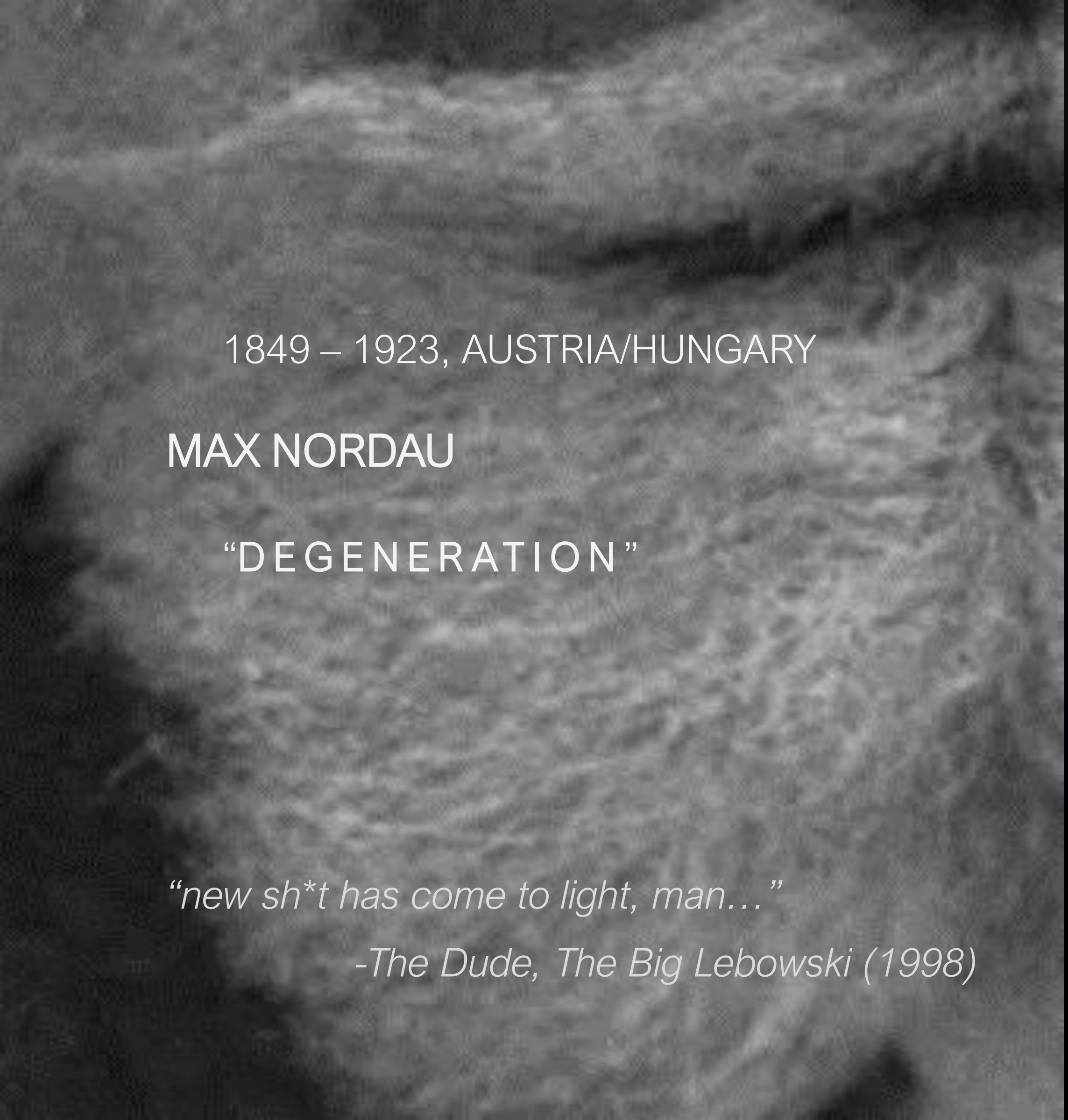
-The Dude, The Big Lebowski (1998)

What’s more **$$$?**

Option 1: Caves>Stay In Caves>Why change?>Abolish all change> Caves are our culture>Caves and combing beards.

Option 2: Caves>What’s that outside>Wait you can do that?>Wow that too?>What else can we do?>Keep looking…

Evolution



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

“DEGENERATION”

*“new sh*t has come to light, man…”*

-The Dude, The Big Lebowski (1998)

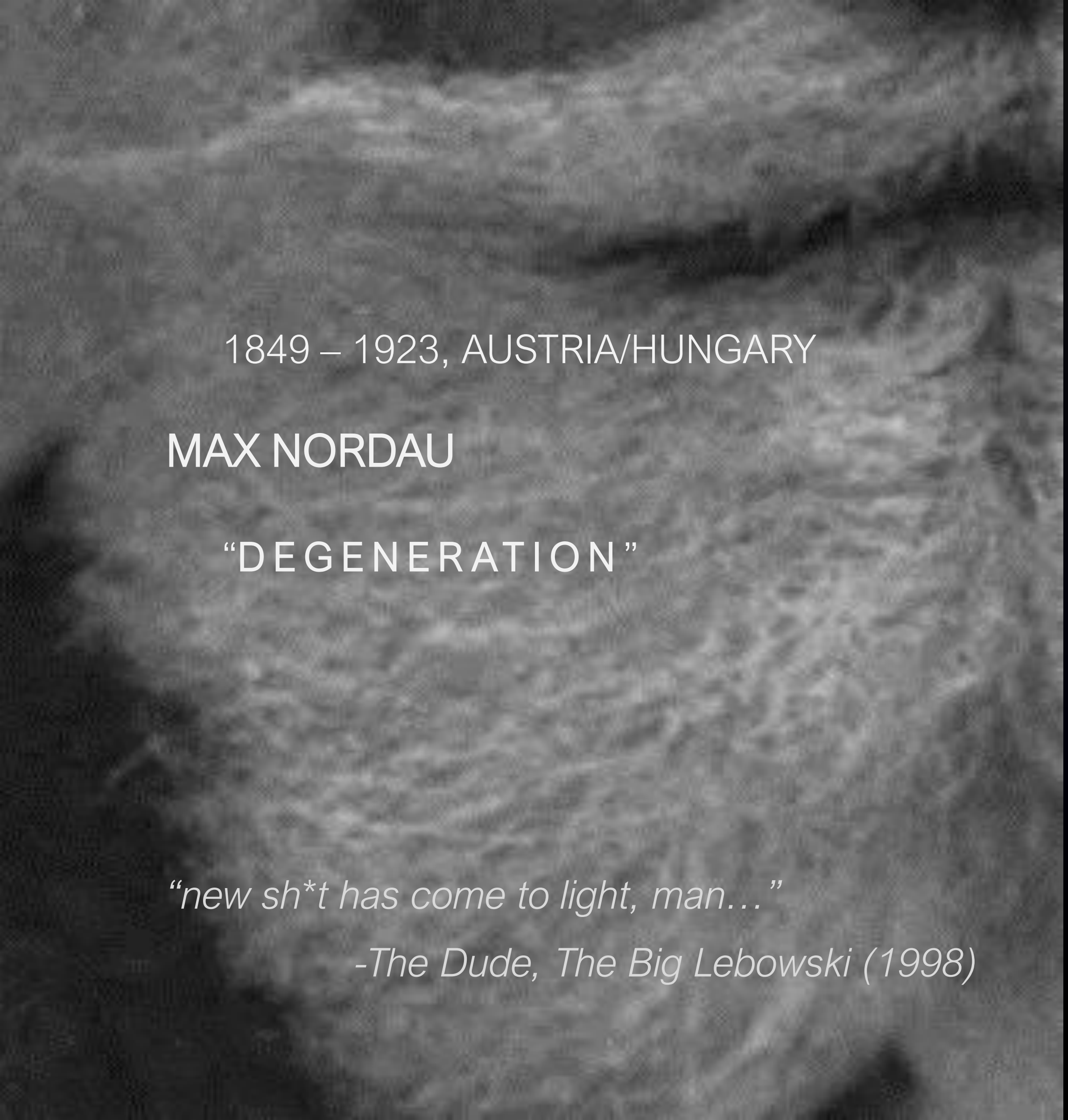
Option **2** obvs!

Our brains are hardwired to seek the “NEW”.

Instagram reels, YouTube, Socials…

Data supports Option 2

Evolution



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

“DEGENERATION”

*“new sh*t has come to light, man…”*

-The Dude, The Big Lebowski (1998)

in seeking

“the glory of the former days”,

Our friend Mr. Chin-Curtain,

lost his ability of

AESTHETIC

TASTE

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION

1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

“DEGENERATION”

Every generation thinks the younger ones’ musical choices are circumspect. It has never been the case.

CC had the intellect, the access and the privilege, but did not have the PERCEPTION to really understand or appreciate the music.

AESTHETIC

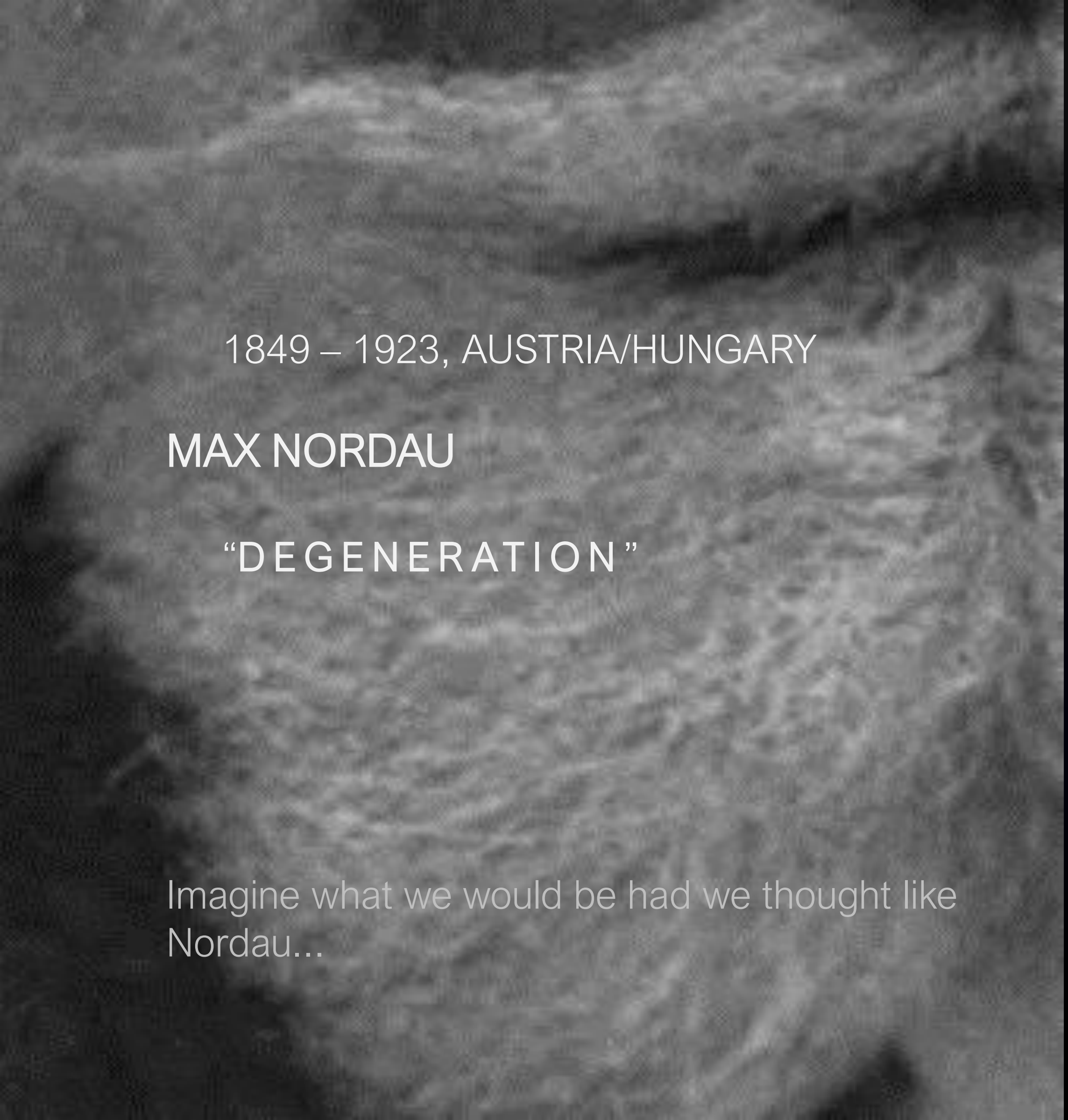
TASTE

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION



1849 – 1923, AUSTRIA/HUNGARY

MAX NORDAU

"DEGENERATION"

Imagine what we would be had we thought like Nordau...

Wagner inspired Vaudeville in US and UK, Parsi Theatre in India

Parsi Theatre inspired songs in our films.

Our films are unique in the world because of Wagner's

AESTHETIC

TASTE

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION

"The unexamined life is not worth living"

- Socrates, 399 BC

TASTE

AESTHETIC

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION

sans evolving perception

we stop enjoying things (new music for e.g.)

we stop caring

ennui

TASTE

AESTHETIC

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION

less bored, less boring – how?

the French have this concept called

RAFFINÉ

TASTE

AESTHETIC

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION

RAFFINÉ

learn a bit more, makes you interested a bit more, and so you learn a bit more and on and on, thus you “refine”

interested === interesting

* notice the triple = sign, if you know you know

TASTE

AESTHETIC

APPRECIATION

DISCERNMENT

perceptiveness

PERCEPTION

RAFFINÉ

learn a bit more, makes you interested a bit more, and so you learn a bit more and on and on, thus you “refine”

interested === interesting

* notice the triple = sign, if you know you know

TASTE

AESTHETIC

APPRECIATION

DISCERNMENT

perceptiveness

‘nuff said

two

we always hear

"Tech's changing so fast, it's hard to keep up..."

love the sea, not the boat

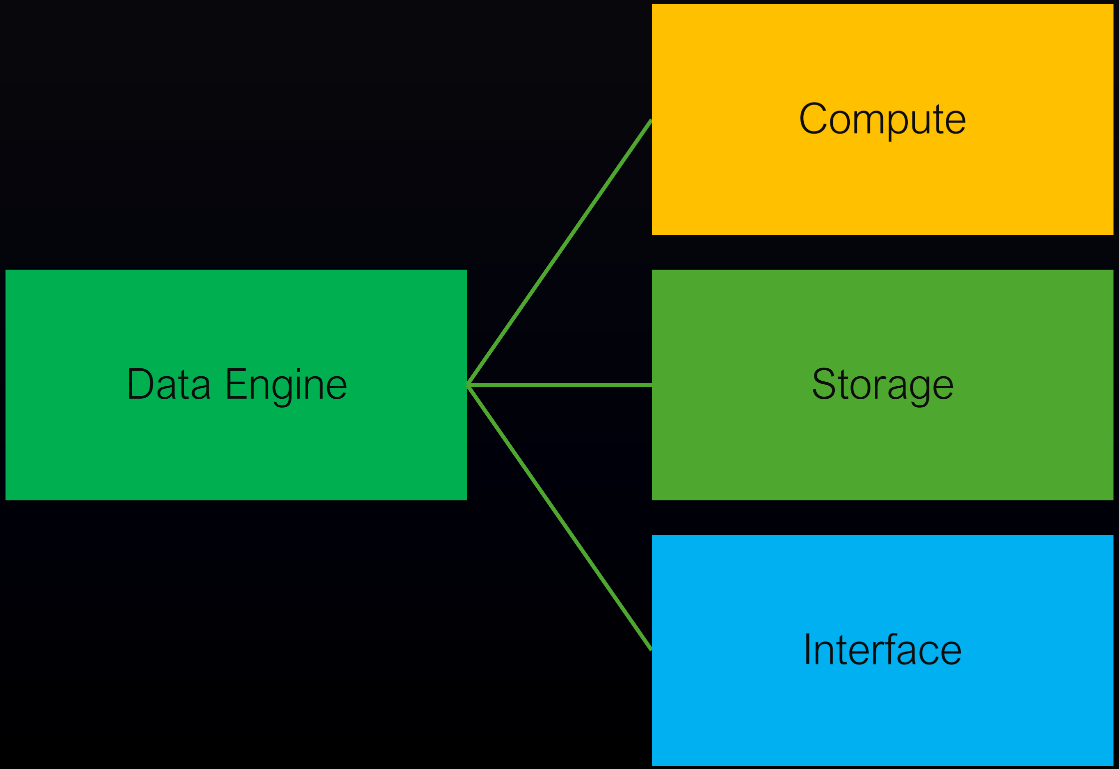
IMO, if you stay interested, it’s never hard let’s see inside an engine

see how it works

how it works

a quick decision tree

decision tree



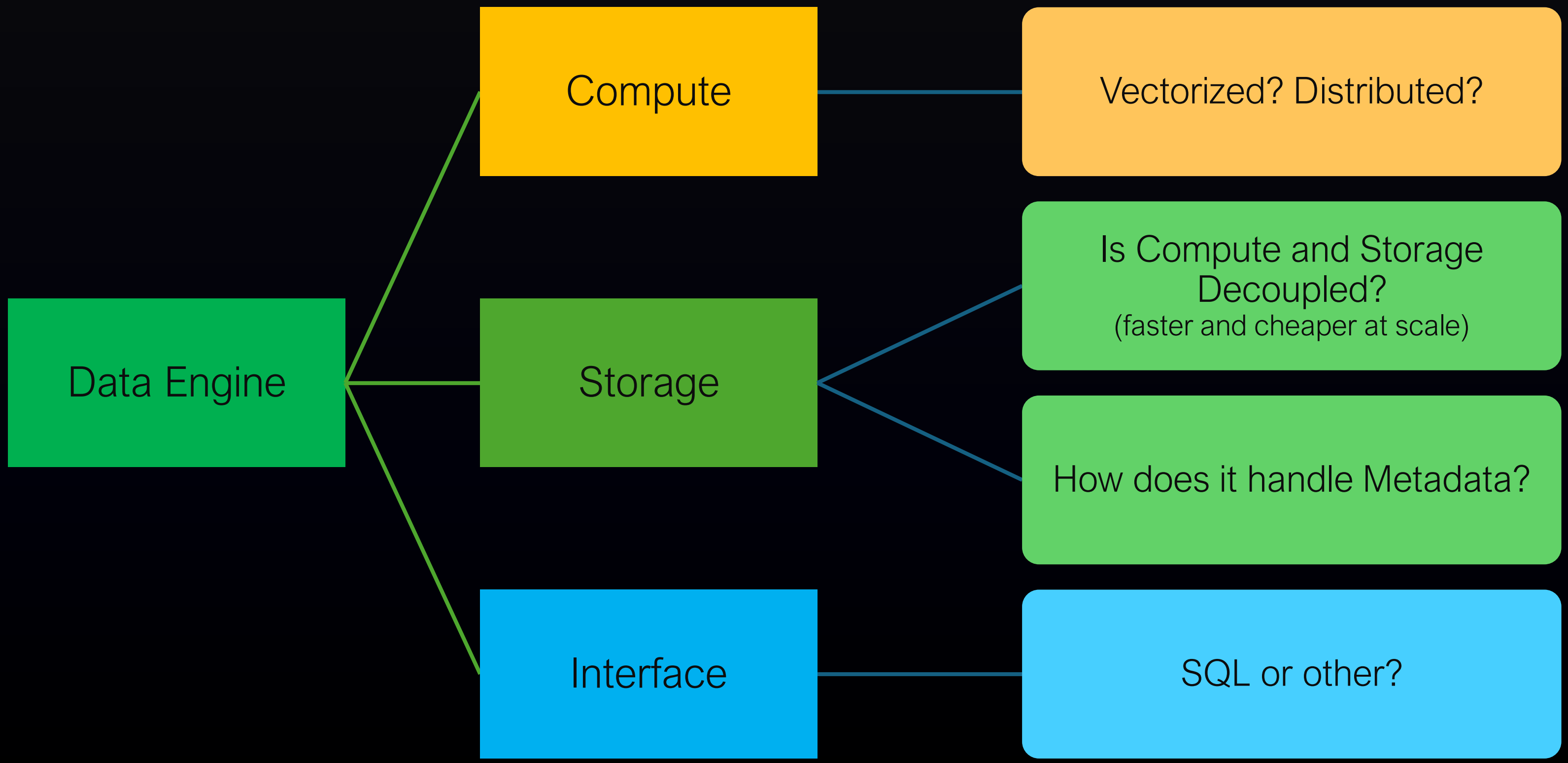
Data Engine

Compute

Storage

Interface

decision tree



Data Engine

Compute

Storage

Interface

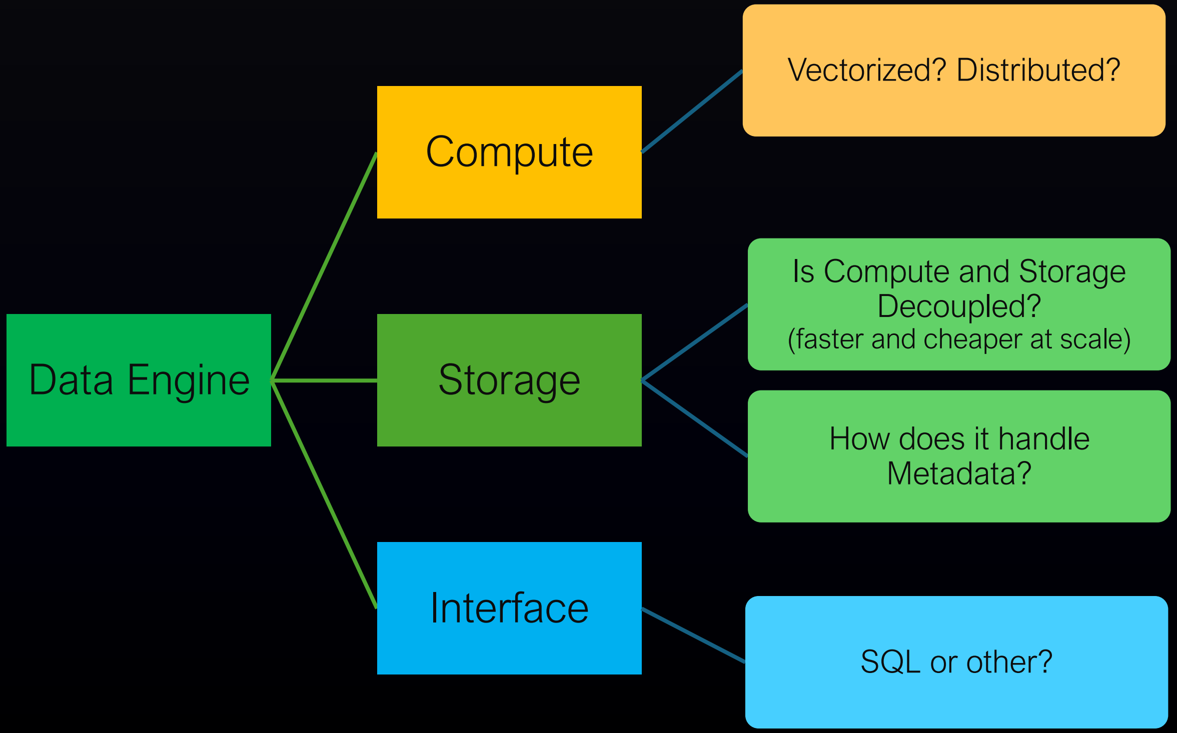
Vectorized? Distributed?

Is Compute and Storage Decoupled?
(faster and cheaper at scale)

How does it handle Metadata?

SQL or other?

decision tree


Data Engine

Compute

Vectorized? Distributed?

Storage

Is Compute and Storage Decoupled?
(faster and cheaper at scale)

How does it handle Metadata?

Interface

SQL or other?

NUMA aware, SIMD (Flynn's classification)

In-memory representation

Separate Store and Compute prove faster and cheaper at scale in most cases

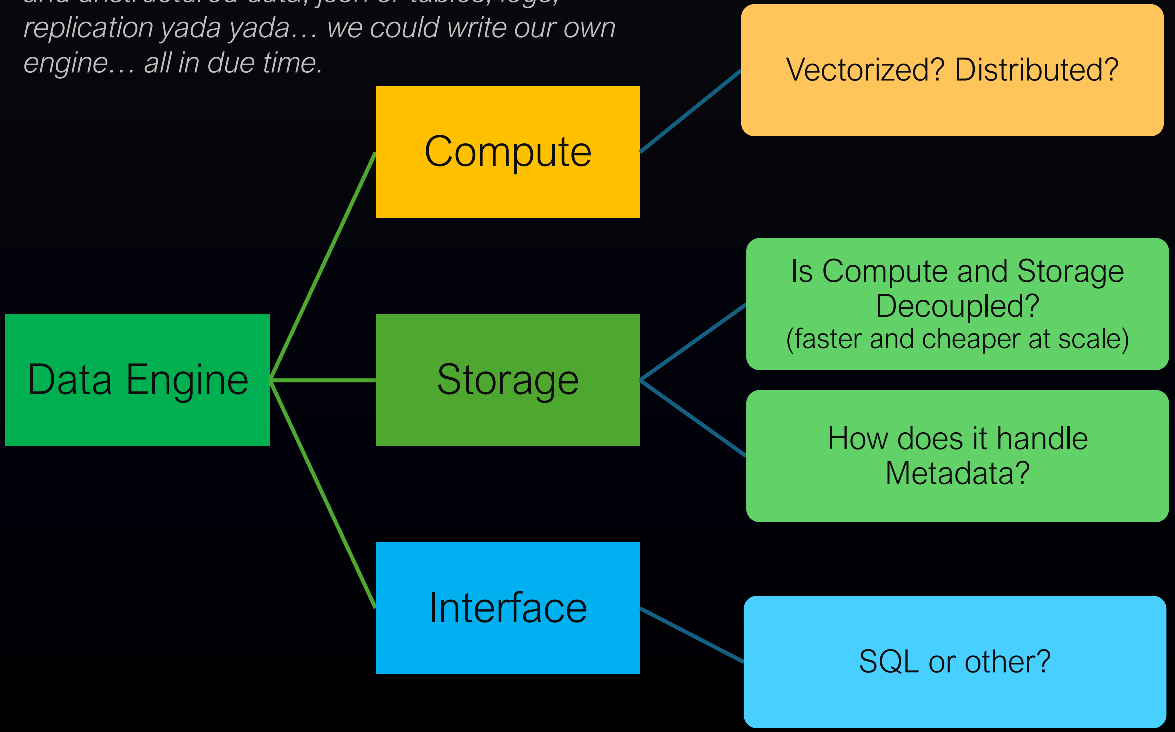
Row-wise or Columnar representation

SQL, PRQL, Substrait, others

JIT compiled? Pre-compiled?

decision tree – breaks the ice, not comprehensive

we could talk about batch vs streaming, structured and unstructured data, json or tables, logs, replication yada yada… we could write our own engine… all in due time.



Data Engine

Compute

Vectorized? Distributed?

Storage

Is Compute and Storage Decoupled?
(faster and cheaper at scale)

How does it handle Metadata?

Interface

SQL or other?

NUMA aware, SIMD (Flynn's classification)

In-memory representation

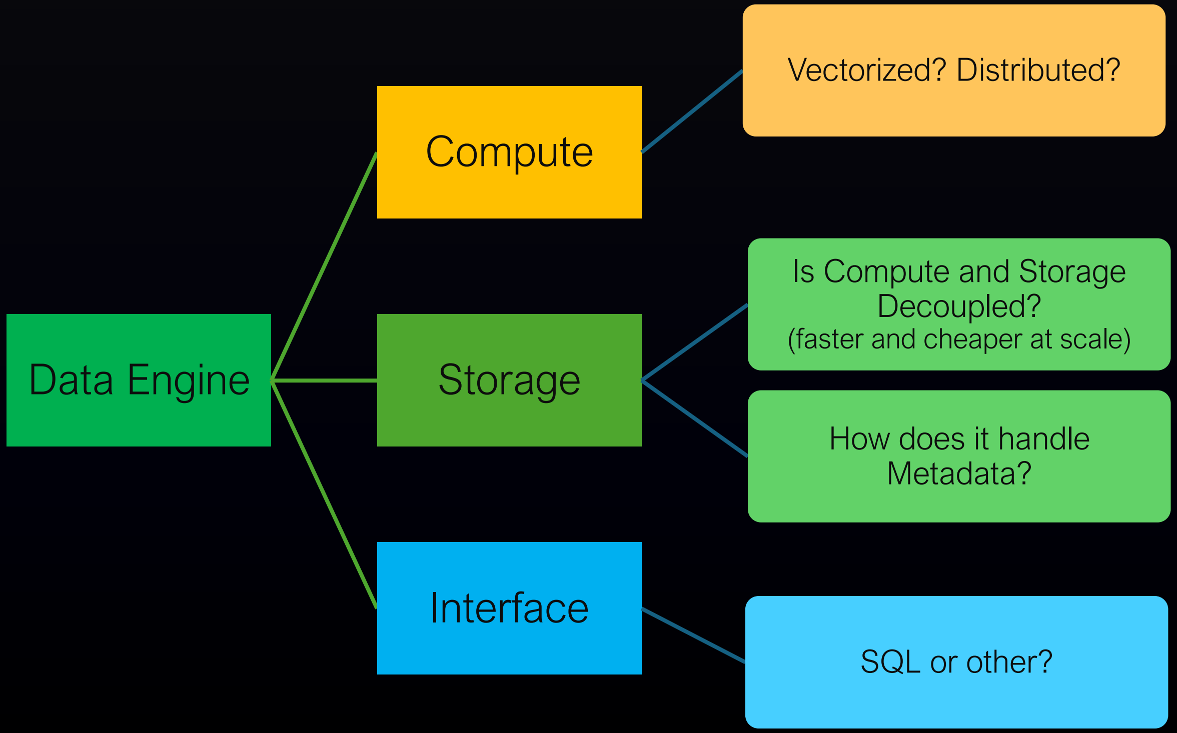
Separate Store and Compute prove faster and cheaper at scale in most cases

Row-wise or Columnar representation

SQL, PRQL, Substrait, others

JIT compiled? Pre-compiled?

Pandas



Data Engine

Compute

Vectorized? Distributed?

Storage

Is Compute and Storage Decoupled?
(faster and cheaper at scale)

How does it handle Metadata?

Interface

SQL or other?

SIMD but not distributed
Row-wise (Numpy) vectorized
Columnar (Arrow) vectorized

Separate Store and Compute – can load data and save data to varied stores

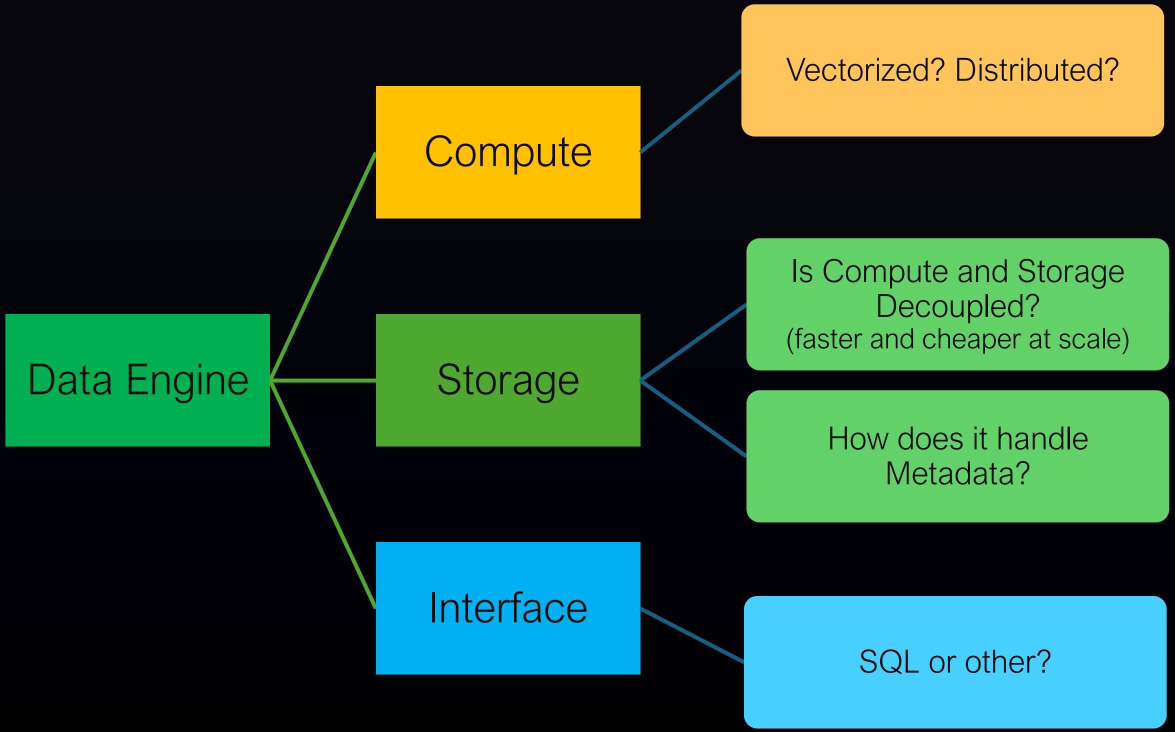
Row-wise (Numpy), new version also has support for Arrow (Columnar) representation

No built-in isolation for metadata

Pandas data frames and indexes

Python based execution

Postgres



Data Engine
Compute
Storage
Interface
Vectorized? Distributed?
Is Compute and Storage Decoupled?
(faster and cheaper at scale)
How does it handle Metadata?
SQL or other?

SIMD but not distributed
Row-wise

Unclear if it’s vectorized yet

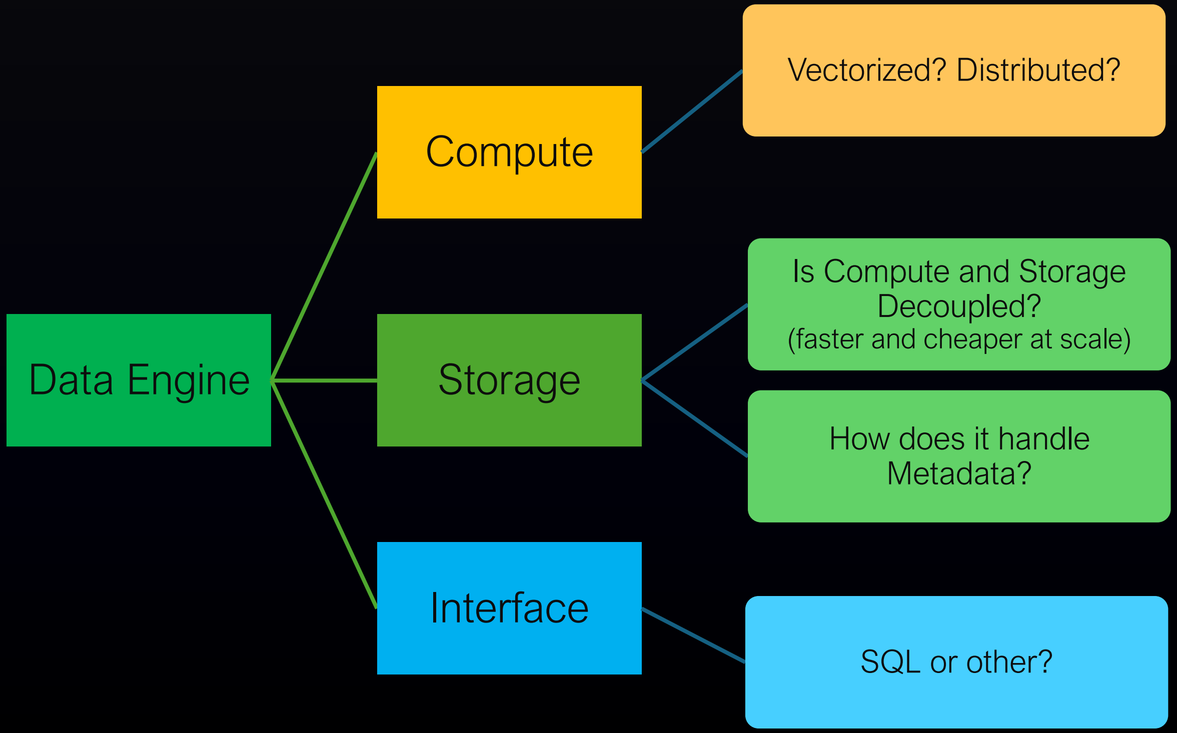
Store and Compute not separate

Row-wise representation

Metadata: `pg_catalog, information_schema`

SQL (specific flavor)

Apache Spark



Data Engine

Compute

Vectorized? Distributed?

Storage

Is Compute and Storage Decoupled?
(faster and cheaper at scale)

How does it handle Metadata?

Interface

SQL or other?

SIMD, distributed
Row-wise
Photon, Velox supports vectorized execution

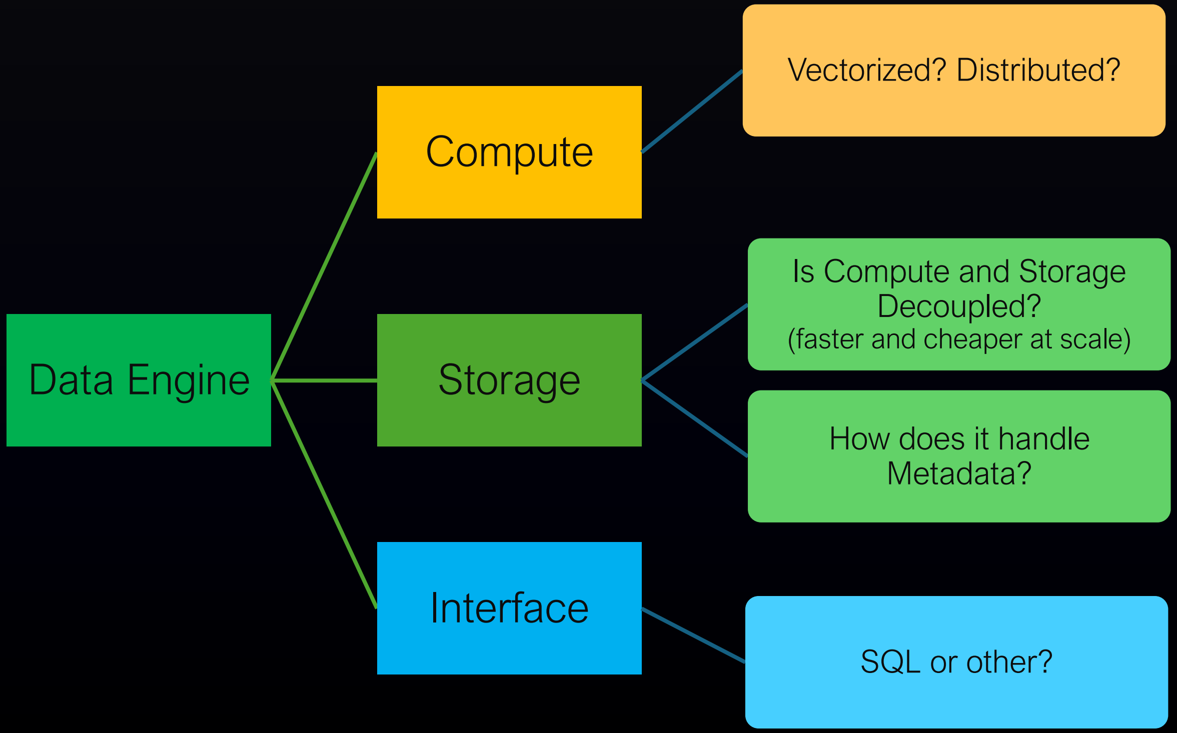
Store and Compute completely separate

Row-wise representation, can handle columnar storage (but not compute) through Parquet, hybrid through Iceberg

Spark Catalogue handles Metadata (who knows about the _spark_metadata directory?) Parquet supports custom metadata.

SparkSQL, Spark Dataframes, RDD

Velox


Data Engine
Compute
Storage
Interface
Vectorized? Distributed?
Is Compute and Storage Decoupled?
(faster and cheaper at scale)
How does it handle Metadata?
SQL or other?

SIMD, distributed
Columnar (?)
Vectorized execution

Just an execution engine, depends on connectors for this stuff, incl. Metadata

in-memory catalog to cache metadata

Connector dependent

highlights - easy rubric

lazy / eager execution

scalar / vector operations

data representation in memory (row-wise / columnar)

data representation on disk (row-wise / columnar)

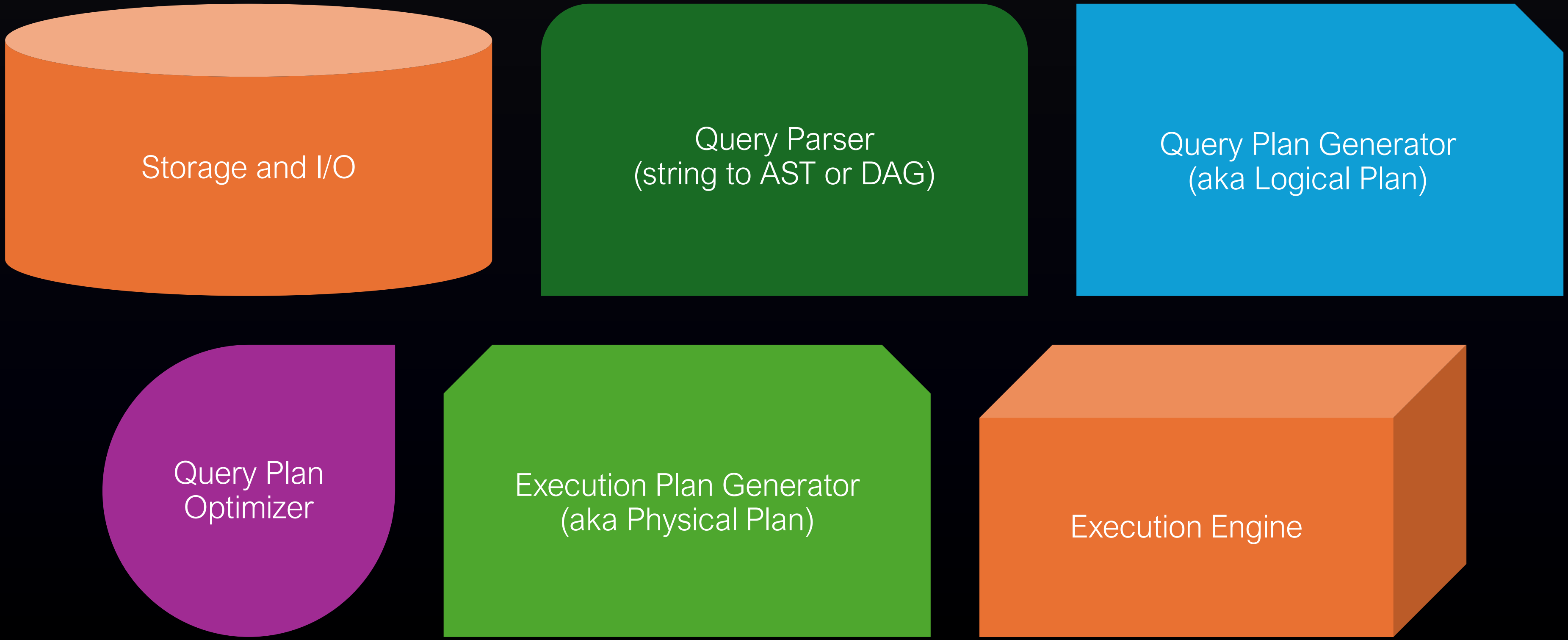
task scheduling, optimization considerations

how 'poly' does the ecosystem get? - underlying complexities

how it works

inside a data engine

what any engine looks like



Storage and I/O

Query Parser
(string to AST or DAG)

Query Plan Generator
(aka Logical Plan)

Query Plan
Optimizer

Execution Plan Generator
(aka Physical Plan)

Execution Engine

what any engine looks like

All engines have some version of these components

Postgres, MySQL, Pandas, Numpy, Spark, Athena, Presto/Trino, Dask, Polars, Cassandra,

DuckDB, Arrow, Velox, Ray, Photon and on and on …

Velox and Photon are just execution engines, but we'll talk about that later.

big 4 out of 6

Tokenizer

- Splits the input query string into a sequence of tokens (keywords, identifiers, operators, literals).

Parser

- Analyzes the token stream to create a structured representation of the query's syntax (Abstract Syntax Tree – AST or Directed Acyclic Graph – DAG)

Query Planner / Optimizer

- Transforms the AST/DAG into an optimized execution plan that outlines how the query will be processed.

Execution Engine

- Executes the optimized physical plan to produce the query results.

big 4 out of 6 - details

Tokenizer

- **tokenize()**: Splits input string into a list of tokens.
- **Keyword Recognition**: Identifies keywords (e.g. SELECT, WHERE, JOIN).
- **Literal Parsing**: Handles numeric, string, and other literal values.

Parser

- **parse()**: Builds the AST or DAG from tokens.
- **Clause Parsers**: Handle specific query clauses (SELECT, FROM, WHERE, GROUP BY, etc.).
- **Grammar Rules**: Defines the valid syntax of the query language.

Query Planner / Optimizer

- **build_plan()**: Initial logical plan from AST.
- **Optimize**: Apply transformations & estimate costs (e.g. predicate pushdown, join reordering).
- **Enumerate**: Explore alternatives for best plan.

Execution Engine

- **Operators**: Perform tasks (e.g., scan, filter, join).
- **Pipeline**: Operators in sequence.
- **Memory**: Manages intermediate results.
- **Parallel**: Coordinates parallel execution.

how it works

representing data in memory (or disk)

row by row representation

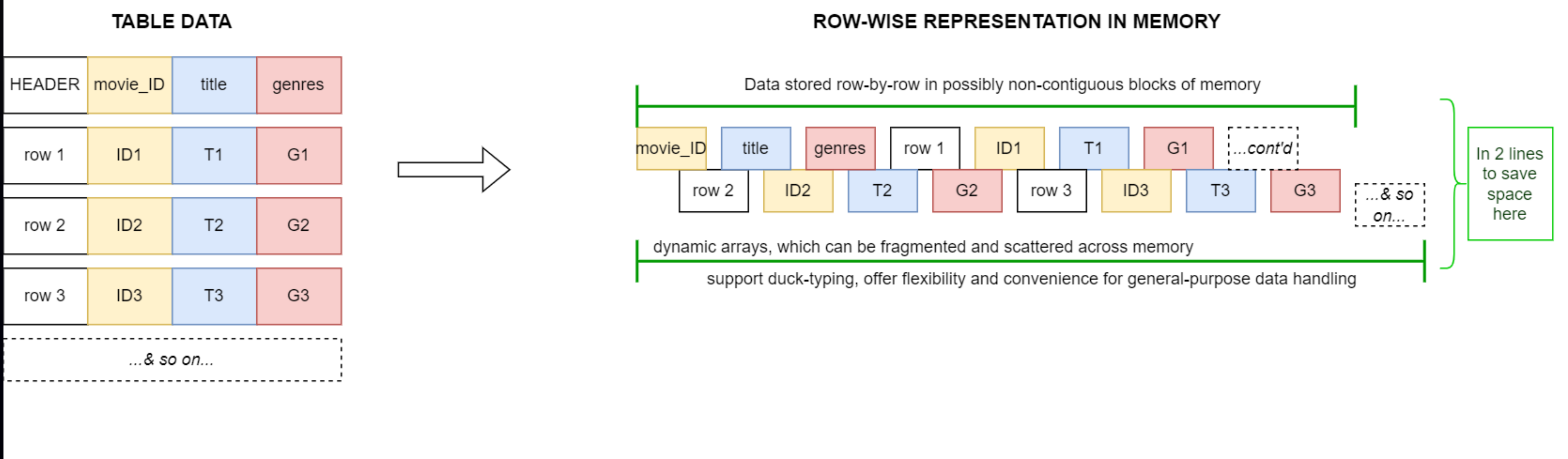


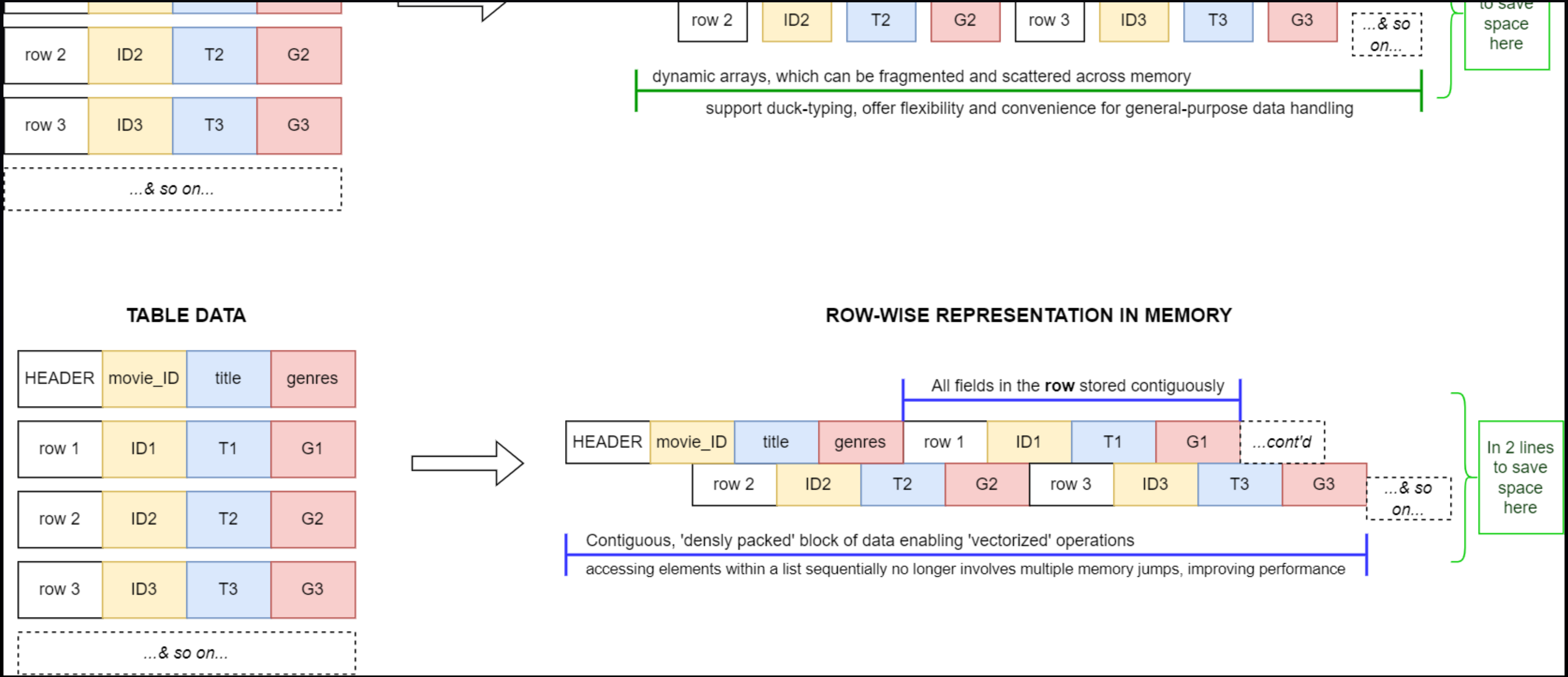
TABLE DATA

HEADER	movie_ID	title	genres
row 1	ID1	T1	G1
row 2	ID2	T2	G2
row 3	ID3	T3	G3
...& so on...			

ROW-WISE REPRESENTATION IN MEMORY

Data stored row-by-row in possibly non-contiguous blocks of memory
movie_ID
title
genres
row 1
ID1
T1
G1
...cont'd
row 2
ID2
T2
G2
row 3
ID3
T3
G3
...& so on...
dynamic arrays, which can be fragmented and scattered across memory
support duck-typing, offer flexibility and convenience for general-purpose data handling
In 2 lines to save space here

contiguous row by row representation



row 2	ID2	T2	G2
row 3	ID3	T3	G3
...& so on...			

row 2	ID2	T2	G2	row 3	ID3	T3	G3	...& so on...

to save space here

dynamic arrays, which can be fragmented and scattered across memory

support duck-typing, offer flexibility and convenience for general-purpose data handling

TABLE DATA

HEADER	movie_ID	title	genres
row 1	ID1	T1	G1
row 2	ID2	T2	G2
row 3	ID3	T3	G3
...& so on...			

ROW-WISE REPRESENTATION IN MEMORY

All fields in the **row** stored contiguously

HEADER	movie_ID	title	genres	row 1	ID1	T1	G1	...cont'd

row 2	ID2	T2	G2	row 3	ID3	T3	G3	...& so on...

In 2 lines to save space here

Contiguous, 'densly packed' block of data enabling 'vectorized' operations

accessing elements within a list sequentially no longer involves multiple memory jumps, improving performance

what's columnar? – *only load the columns you need - PING*

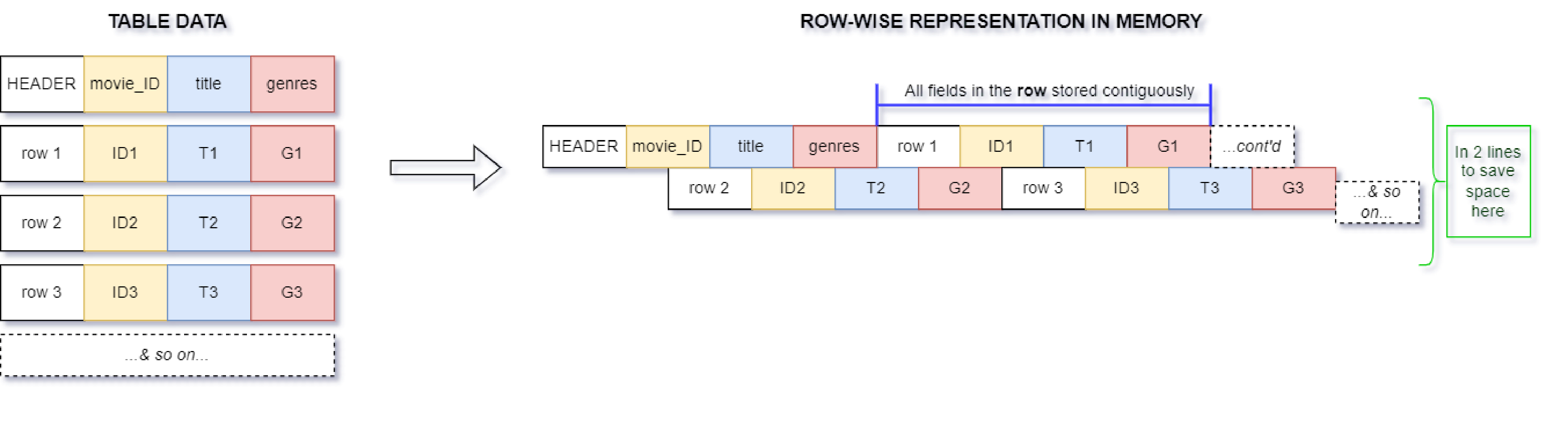


TABLE DATA

HEADER	movie_ID	title	genres
row 1	ID1	T1	G1
row 2	ID2	T2	G2
row 3	ID3	T3	G3
...& so on...			

ROW-WISE REPRESENTATION IN MEMORY

All fields in the **row** stored contiguously

HEADER | movie_ID | title | genres | row 1 | ID1 | T1 | G1 | *...cont'd*

row 2 | ID2 | T2 | G2 | row 3 | ID3 | T3 | G3 | *...& so on...*

In 2 lines to save space here

what’s columnar? – *only load the columns you need - PONG*

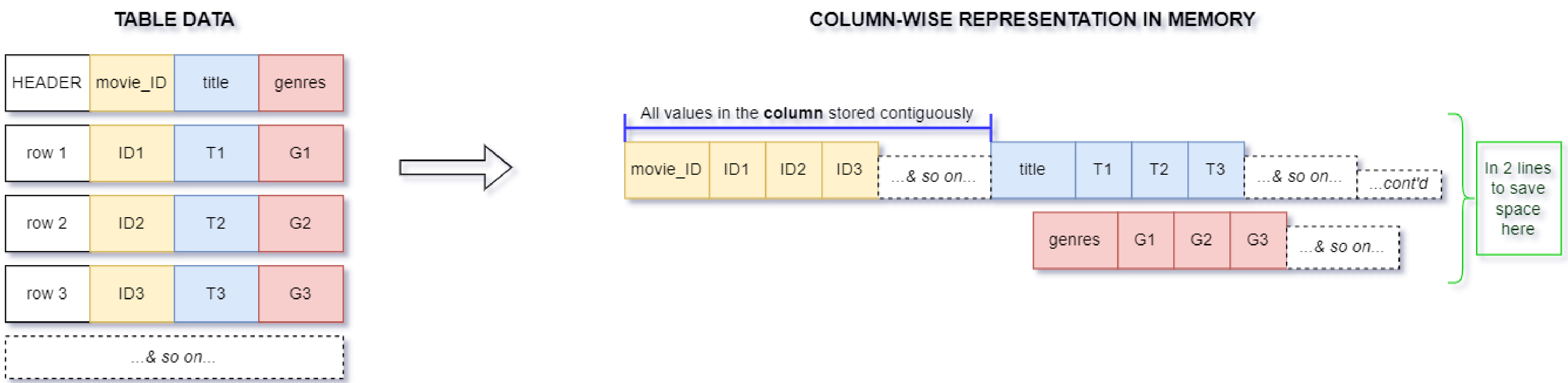


TABLE DATA

HEADER	movie_ID	title	genres
row 1	ID1	T1	G1
row 2	ID2	T2	G2
row 3	ID3	T3	G3

...& so on...

COLUMN-WISE REPRESENTATION IN MEMORY

All values in the **column** stored contiguously

movie_ID	ID1	ID2	ID3	*...& so on...*	title	T1	T2	T3	*...& so on...*	*...cont'd*

genres	G1	G2	G3	*...& so on...*

In 2 lines to save space here

wait a minute…

all that means we can kinda have…

that means we can kinda have…

A framework to build our own data engine

and evaluate other engines…

is that really it?

yes! *

*let's look at some code