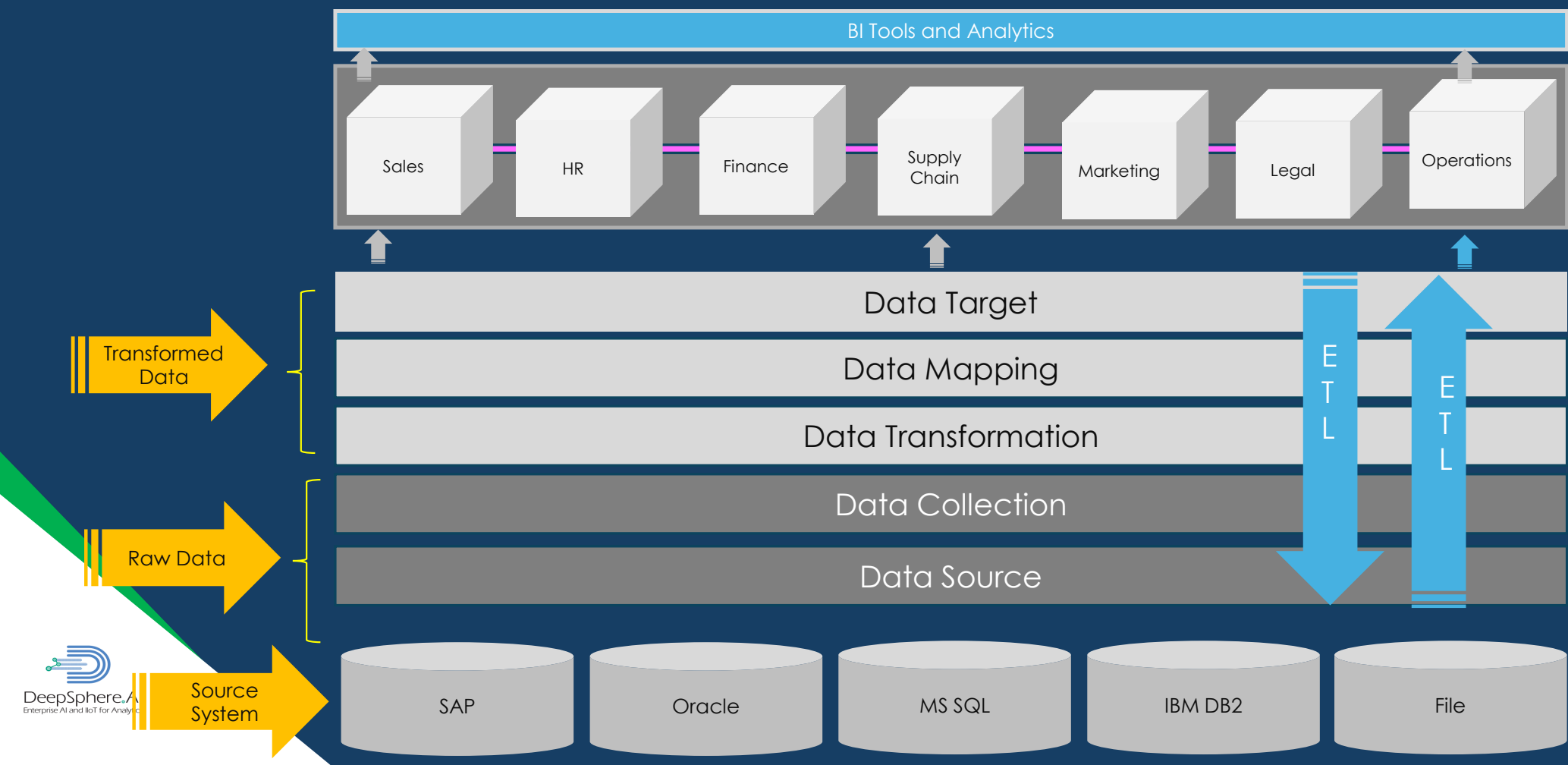
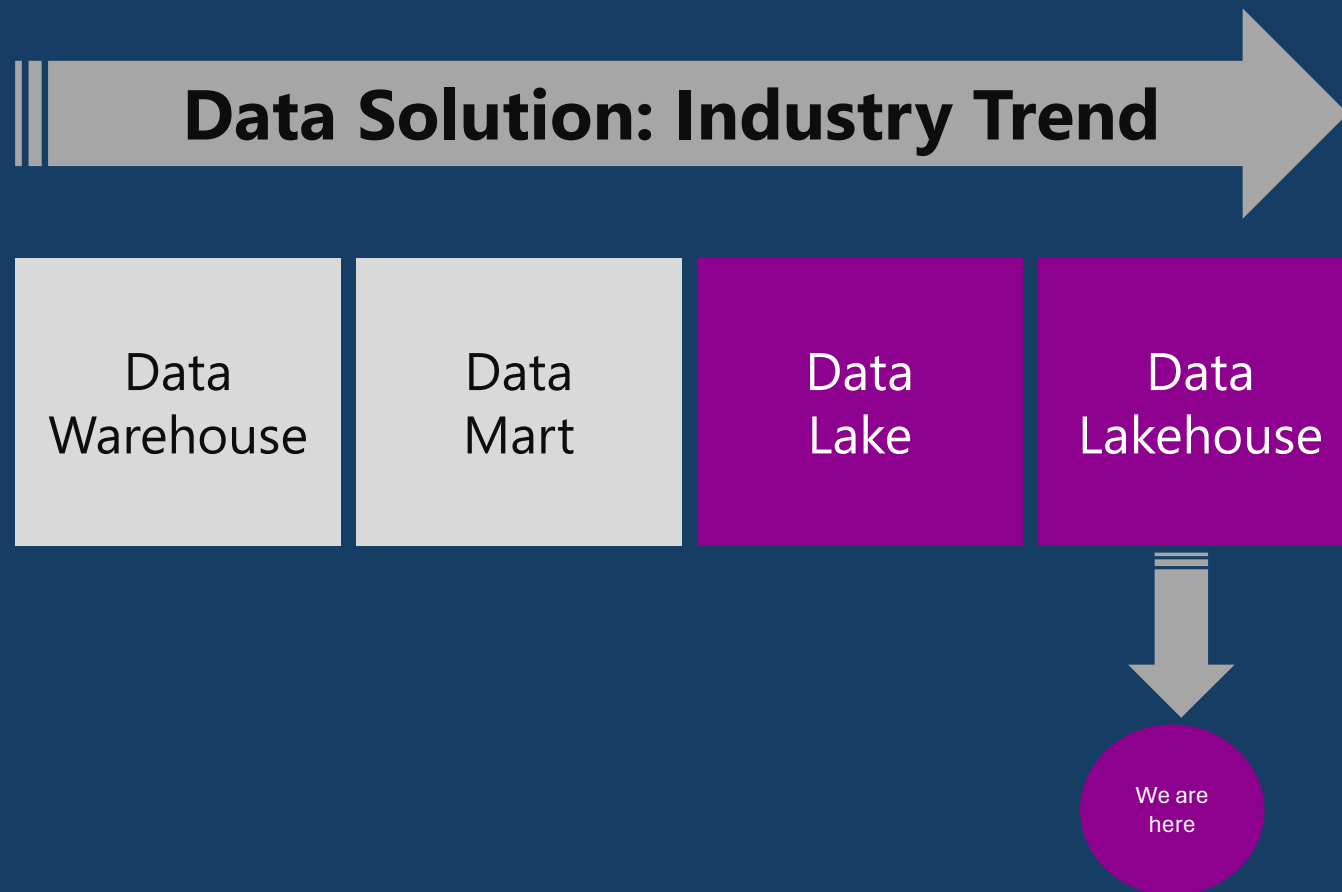


- ✓ **Data Warehousing and Analytics**
- ✓ **Transaction Management**
 - ✓ **ACID properties**
- ✓ **Indexing for Performance:**
- ✓ **Relationships and Constraints:**
- ✓ **Normalization Process:**

Data Warehousing and Analytics



Data Warehousing and Analytics



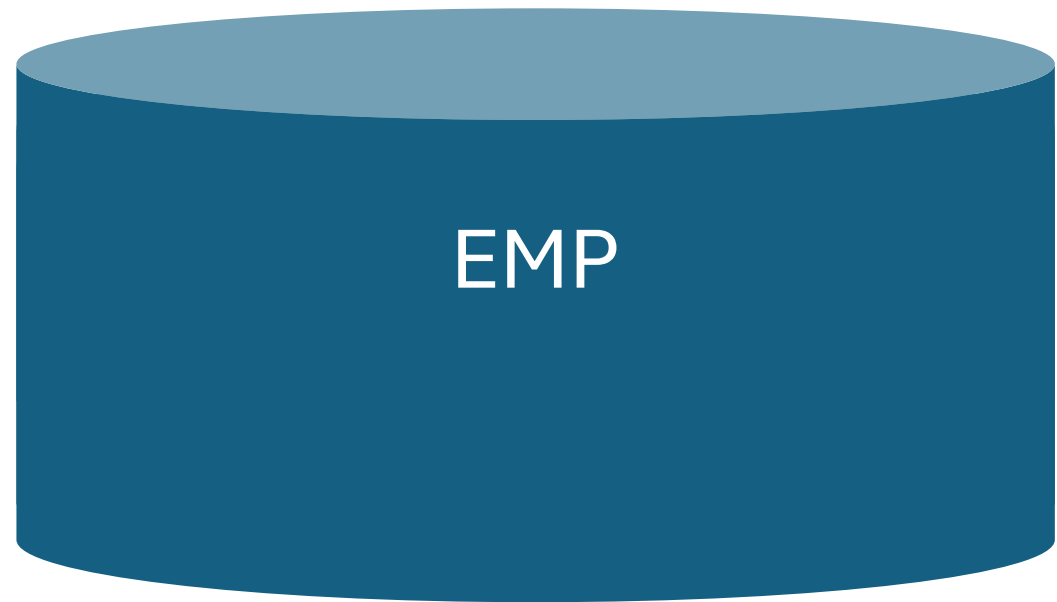
A.C.I.D. properties:

- ✓ **Atomicity**
- ✓ **Consistency**
- ✓ **Isolation**
- ✓ **Durability**

1.Normalization Process:

- ✓ **1NF (First Normal Form):** Ensures that the database table is organized such that each column contains atomic (indivisible) values, and each record is unique. This eliminates repeating groups, thereby structuring data into tables and columns.
- ✓ **2NF (Second Normal Form):** Builds on 1NF by We need to remove redundant data from a table that is being applied to multiple rows. and placing them in separate tables. It requires all non-key attributes to be fully functional on the primary key.
- ✓ **3NF (Third Normal Form):** Extends 2NF by ensuring that all non-key attributes are not only fully functional on the primary key but also independent of each other. This eliminates transitive dependency.

- Select * from EMP where EMPID=101; - The EMPID is not indexed, the time it takes to retrieve the record is 2 Min
- Create index JP_INDEX on EMP (EMPID)
- Select * from EMP where EMPID=101; the time it takes to retrieve the record is 1 Min



Customer
Table

Product
Table

Relationship

- 1: 1
- 1: Many
- Many : Many

Python doesn't have the concept called private variables. However, most of the Python developers follow a naming convention to tell that a variable is not public and it's private. We have to start a variable name with a double underscore to represent it as a private variable (not really).

Convention	Example	Meaning
Single leading underscore	<code>_variable</code>	Indicates that the name is meant for internal use only
Single trailing underscore	<code>class_</code>	Avoids naming conflicts with Python keywords and built-in names
Double leading underscore	<code>__attribute</code>	Triggers name mangling in the context of Python classes
Double leading and trailing underscore	<code>__name__</code>	Indicates special attributes and methods that Python provides
Single underscore	<code>_</code>	Indicates a temporary or throwaway variable

Constraints

- ✓ **Primary Key**
- ✓ **Unique**
- ✓ **Not Null**
- ✓ **Default**
- ✓ **Check**

- **Constraints it ensures the data integrity in the database**

When to use what, and what is the difference between each other

Data Structure
in Python

List

[

Tuple

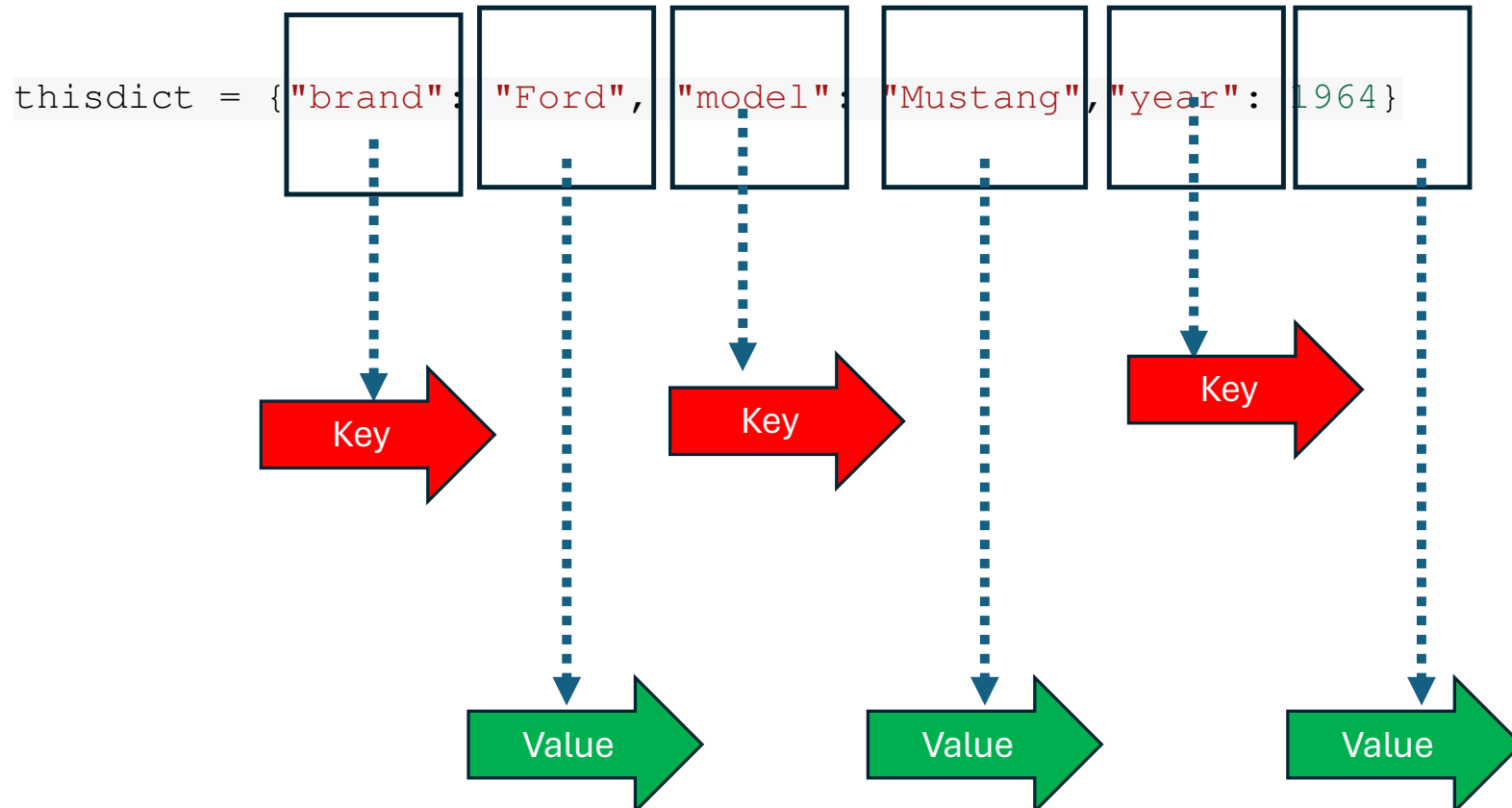
(

Set

{

Dictionary

{



- In dictionary the data organized in the KEY and VALUE format, we need to use the KEY to access the values
- Dictionary will take duplicate keys, and it will retain only the last values in case if there is any duplicate

```
thisdict = { "brand": "Ford", "electric": False, "year": 1964, "color": ["red", "white", "blue"] }
```

In this case, the KEY has more than one value in the form list, or it could be a set, Tuple or another Dictionary

Color = KEY

A list of Dictionary, in a List, we can have multiple DICTIONARY

```
ls_dict = [{'py': 'Python', 'mat': 'MATLAB', 'cs': 'Csharp'}, {'A': 65, 'B': 66, 'C': 67}, {'a': 97, 'b': 98, 'c': 99}]
```

List

List

A list of Dictionary, in a List, we can have multiple DICT

```
ls_dict = [{'py': 'Python', 'mat': 'MATLAB', 'cs': 'Csharp'}, {'A': 65, 'B': 66, 'C': 67}, {'a': 97, 'b': 98, 'c': 99}]
```

DICT 1

DICT 1

DICT 2

DICT 2

DICT 3

DICT 3