BOSSCODER ACADEMY

SQL vs NoSQL DATABASES

What exactly is the difference?

BOSSCODER ACADEMY

#1

WHAT IS A DATABASE?

A database is a **systematic collection of data** which can be in **different formats.**

Databases are used to **support electronic storage** and for **efficient manipulation** and **management of data.**

BOSSCODER ACADEMY

For Example

Banking system.

The database needs to store, manipulate, and present data related to users, their identification details, their accounts, their activities such as transactions, withdrawals, transfers.

It also needs to keep track of the loans applied by users, different policies offered by the bank and the changing features of them. Apart from users, data of staff, managers and branches also has to be stored and managed.

BOSSCODER ACADEMY

#2

WHAT IS A RELATIONAL DATABASE ?

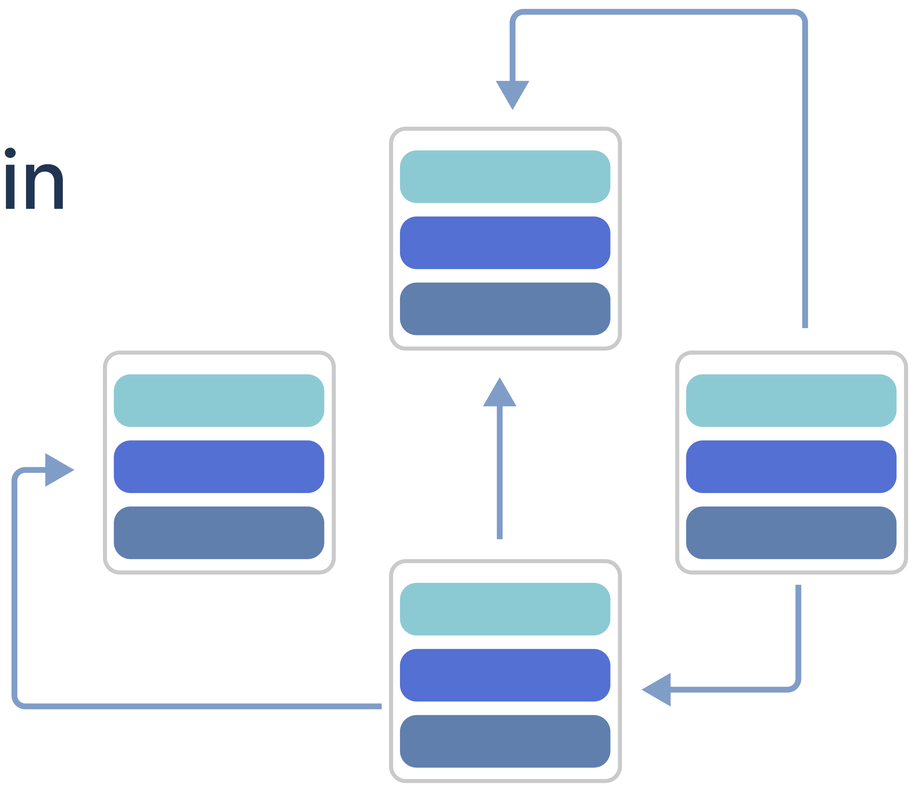
SQL pronounced as “S-Q-L” or as “See-Quel” is primarily **Relational Databases.**

A relational database stores and provides access to data points that are related to one another.

They are based on the relational model,which represents data in tables.

The columns of the table hold attributes of the data. Each row of a relational database represents a record. This makes it easy to establish the relationships among data points or records.

In a relational database, each row in the table is a record with a unique ID called the key.



BOSSCODER ACADEMY

For Example

A simple database for a small shop can have two tables, customer and items.

Customer table will have columns such as ID, Name, Mobile Number, Address, Pincode.

User id	Name	Mobile no.	Address	Pincode
101	Ram	818192953	Sector 9 Plot b 5	500019
102	Shyam	958192953	Krishna Nagar b2	500156
103	Naman	858622953	Gandhi Nagar b10	502513

Similarly Items table can have columns such as ID, Name, Stock, Size, Color

Item id	Name	Stock	Size(cm)	Color
510123	Toothbrush	50 Pcs	10 cm	Blue
510124	Notebook	150 PCs	30 CM	White
510125	Soap	35 PCs	25 CM	Green

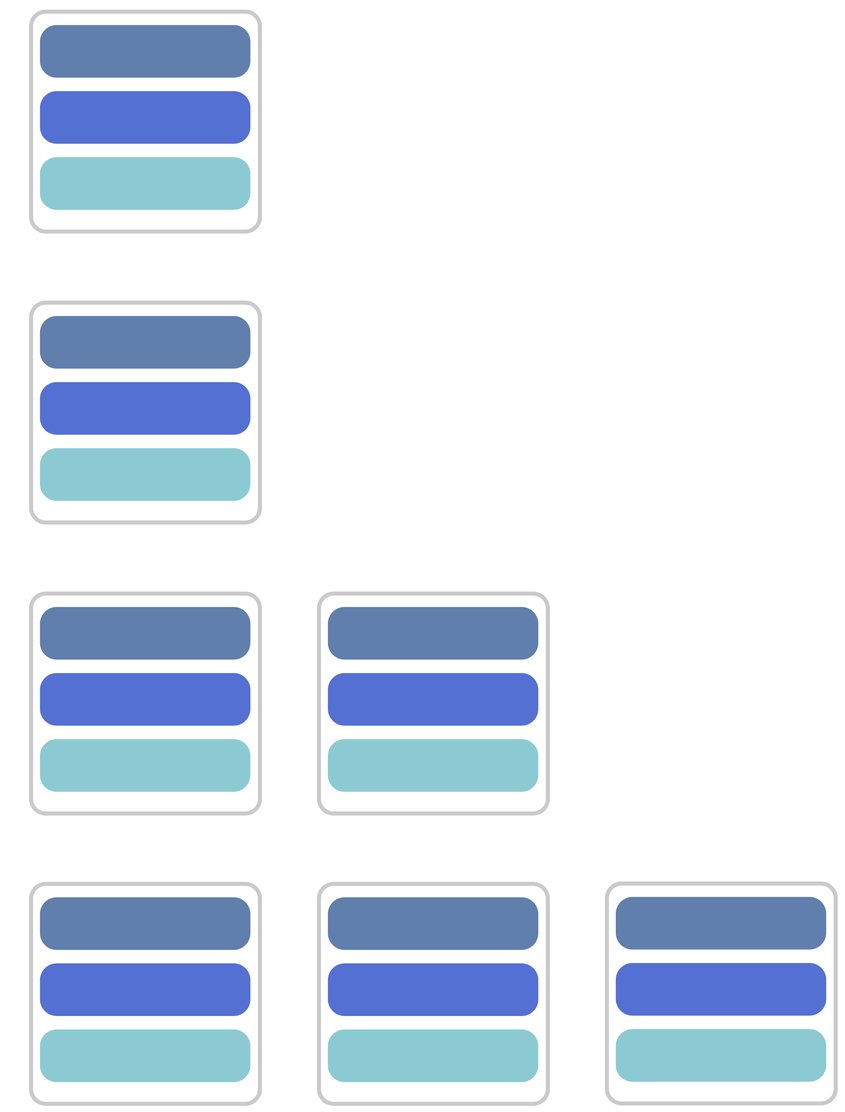
BOSSCODER ACADEMY

NO SQL is a

NON - RELATIONAL OR DISTRIBUTED DATABASE ?

Non-relational databases store their data in a **non-tabular form.**

Instead, non-relational databases have different storage models based on specific requirements of the type of data being stored.



BOSSCODER ACADEMY

For Example

Document databases

Data is stored as documents in a format such as JSON or XML.Each document assigned its own unique key

Left square brackets defines the beginning of JSON text

```
[
{
"empid" :101,
"lastname" : "Sharma",
"firstname" : "Rahul"
"title" : "Programmer" ,
"birthdate": "12-09-1997"
“hiredate" :”18-06-2010”
"address” : ”Sector 09, Chandigarh”,
“Region” : “Punjab”,
“postalcode”: “500025”,
“Country”: “India”
“phone” : “+91 8768561213
”
},
```

“Empid ” is the name (column, 1 is the value (for this row)

Colon depicts assignment of a value to a name

Left and right curly Brackets enclose a JSON Object

Comma seperate this first object from the next JSON object.

BOSSCODER ACADEMY

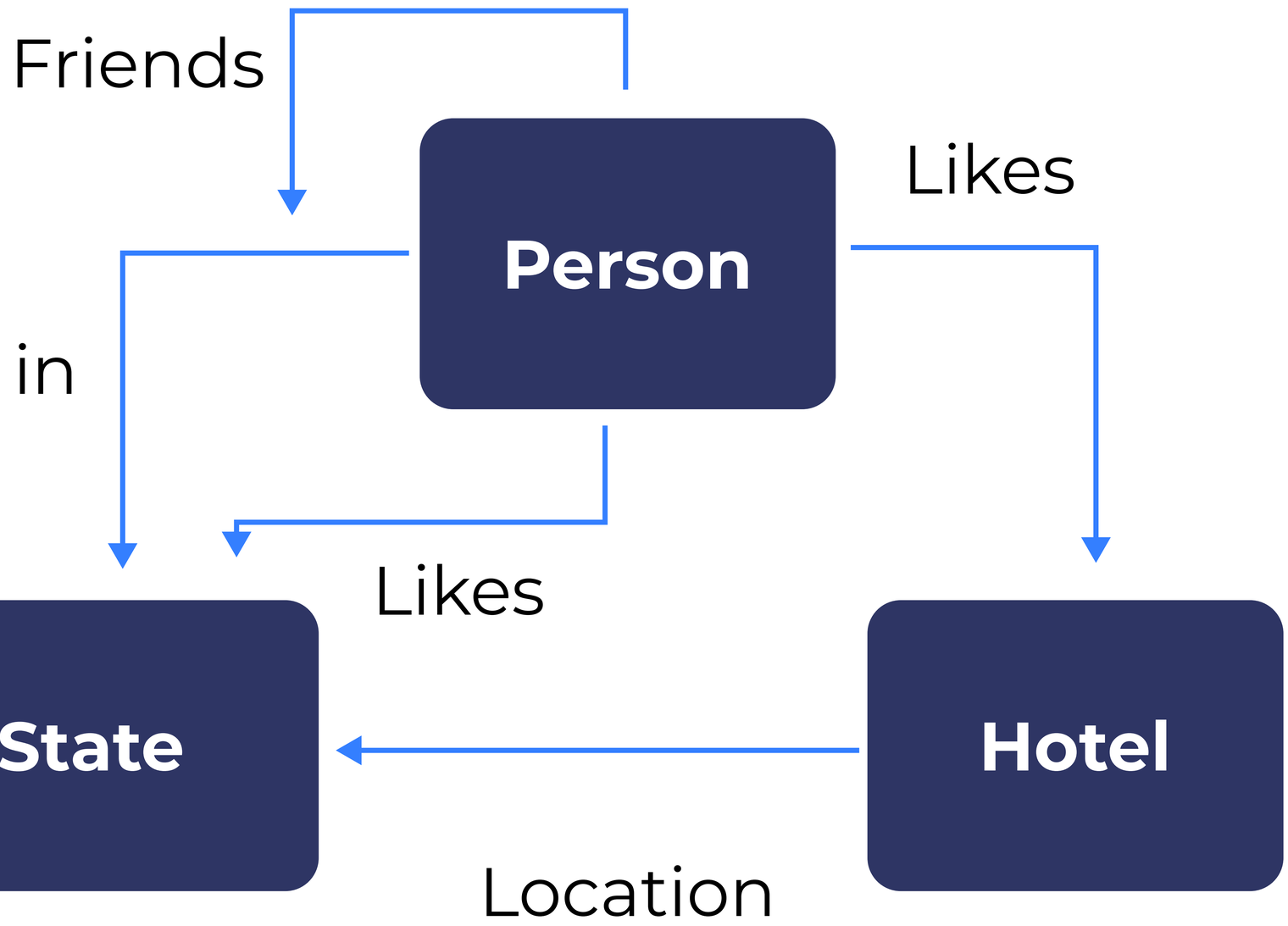
Graphs with nodes and edges

Each element is stored as a node. It stores the data itself.

For Example

A person in a social media graph.

The Edge explains the relationship between two nodes. Edges can also have their own pieces of information, like relationship between two nodes.



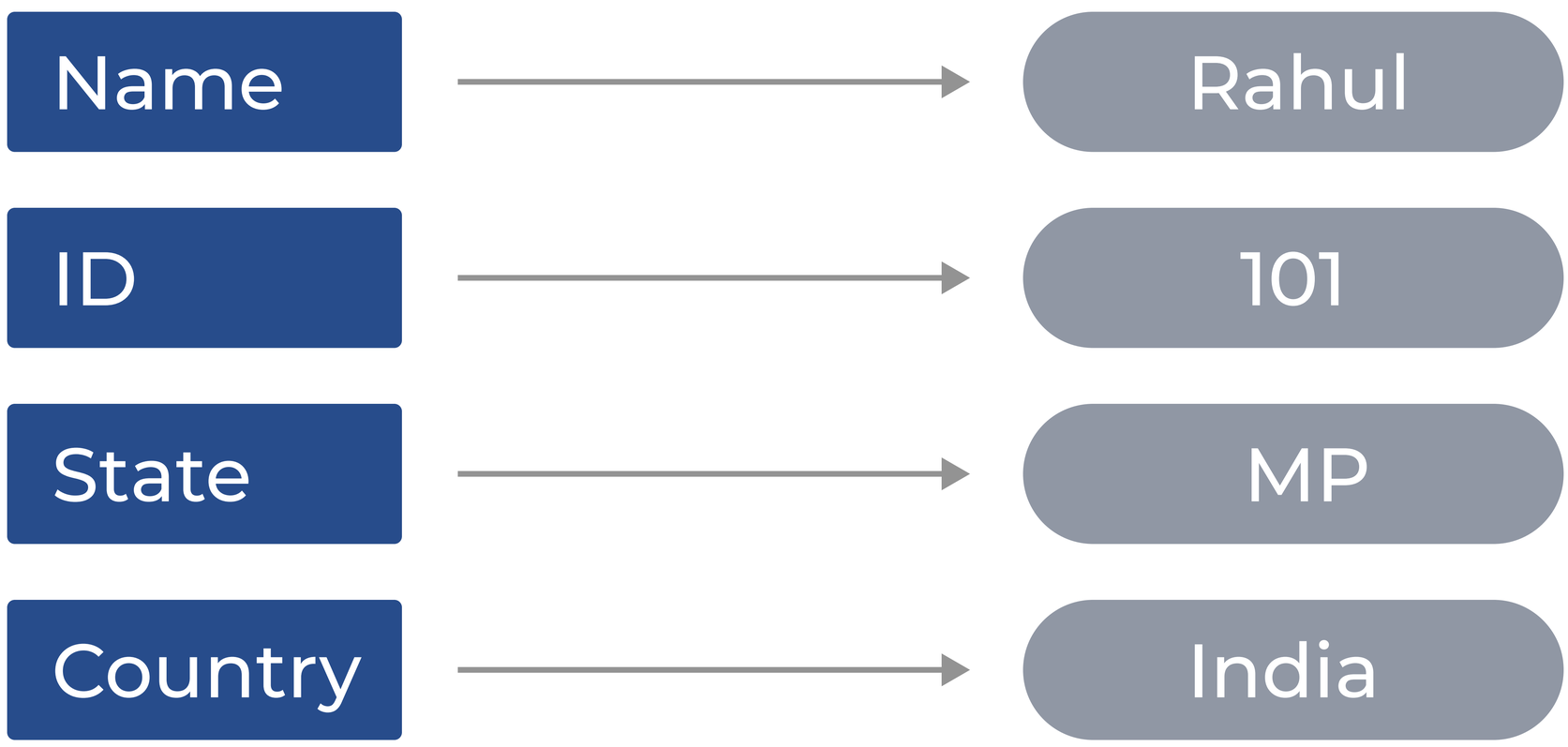
BOSSCODER ACADEMY

Key Value Data-Model

In this model every data element in the database is stored as a key value pair.

The pair consists of an attribute or “key” and its corresponding value.

We can sort of consider this model to be similar to a relational database with only two columns, the key and and the value.



BOSSCODER ACADEMY

Column Oriented Databases

A column oriented database is organised as a set of columns.

Column-oriented storage is used to improve analytic query performance. It reduces the overall disk I/O requirements and reduces the amount of data you need to load from disk.

User id	First Name	Last name	Salary
510123	Rohit	Ranjan	45,000rs
510124	Karan	Singh	40,000rs
510125	Abhay	Kumar	25,000rs

DATA SET

User id	510123	510124	510125
First Name	Rohit	Karan	Abhay
Last name	Ranjan	Singh	Kumar
Salary	45,000rs	40,000rs	25,000rs

Column Oriented Databases

User id	First Name	Last name	Salary
510123	Rohit	Ranjan	45,000rs
510124	Karan	Singh	40,000rs
510125	Abhay	Kumar	25,000rs

Row Oriented Databases

BOSSCODER ACADEMY

Properties followed by

SQL TRANSACTIONS

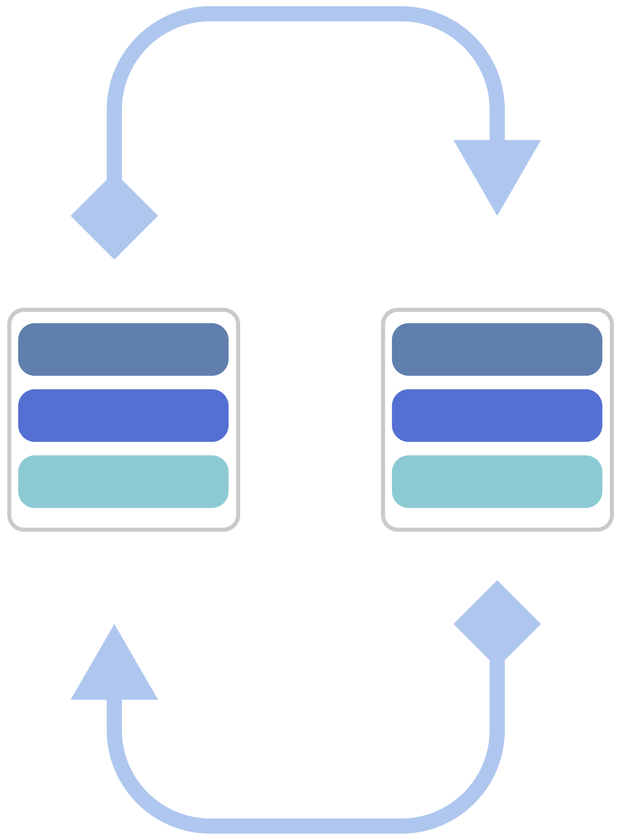
Relational databases follow ACID properties.
ACID in DBMS stands for Atomicity, Consistency, Isolation, and Durability.

Atomicity:

A transaction is a single unit of operation. Any transaction has to be either executed completely or will not be executed at all.

For Example

During a transaction of transfer of funds, if money is deducted from one account but not deposited into another, due to failure of transaction in between, the entire transaction is not executed.



BOSSCODER ACADEMY

Consistency:

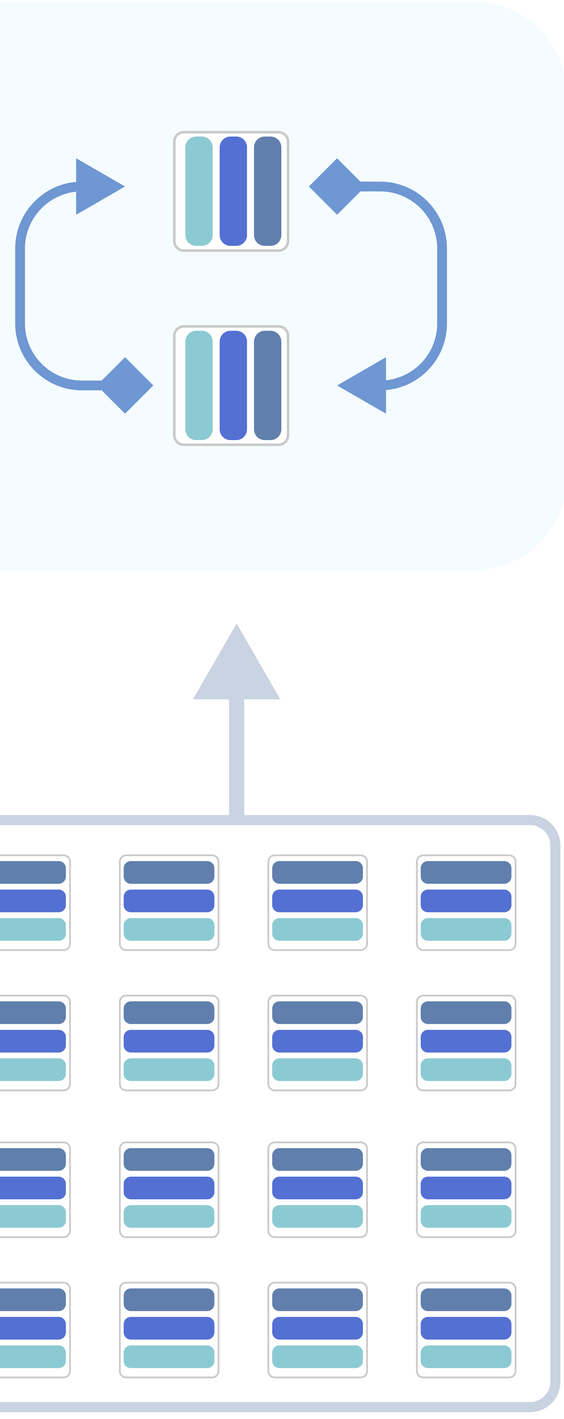
The state of the system before and after transactions should remain consistent.

For Example

During a transaction of transfer of funds, the total money present in the system should be the same.

Isolation:

Every transaction should be executed in isolation from other transactions. If transactions make changes to the database concurrently, it may lead to errors and overwritten data.



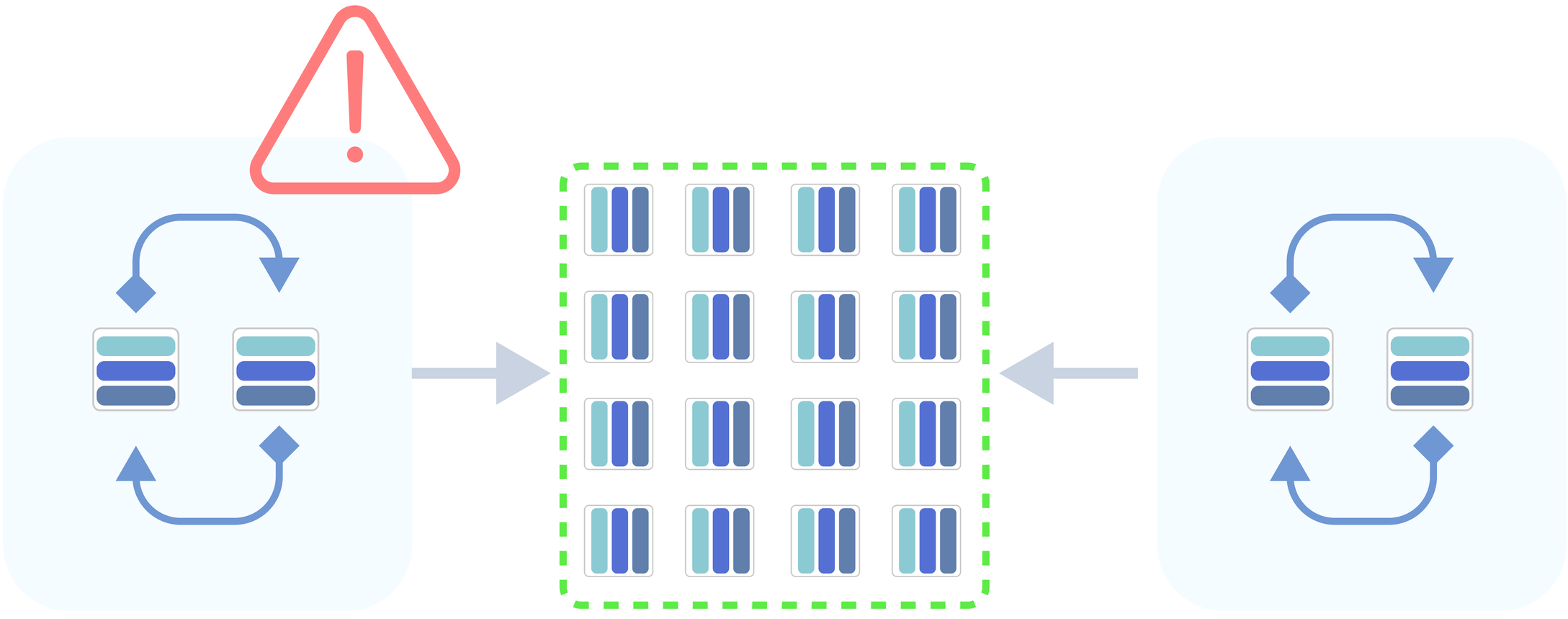
BOSSCODER ACADEMY

For Example

If there are two transactions, one for transfer of funds and other for printing total data, if printing takes place before complete transfer, it will show inconsistent. This is avoided by serializability and locking protocols.

Durability:

After a transaction is successfully completed the changes in the database should persist even if there are any software or hardware failures.



BOSSCODER ACADEMY

What is

CAP THEOREM ?

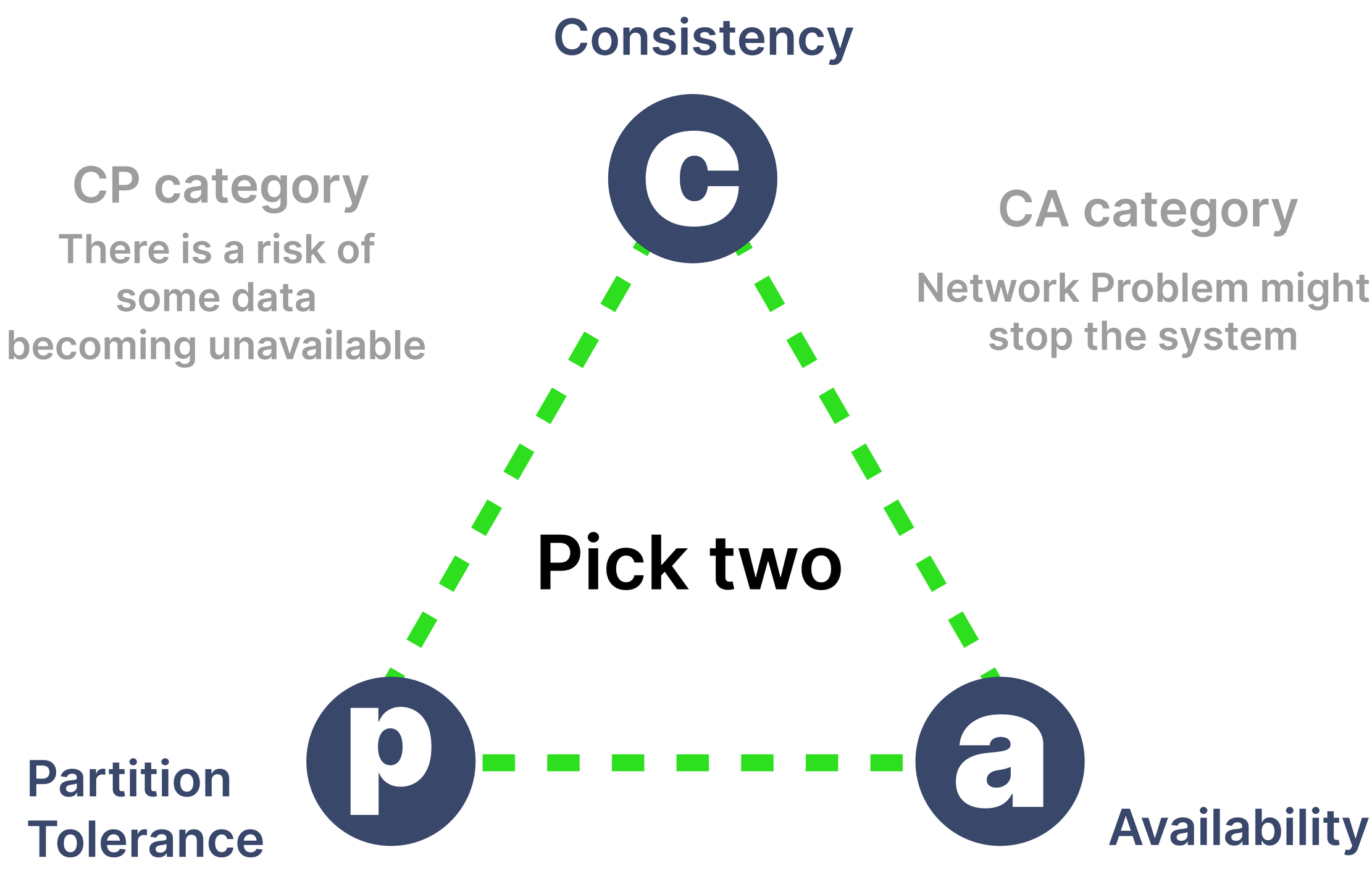
The CAP theorem (Brewer’s theorem) states that a distributed system or database can provide only two out of the following three properties:

- **Consistency:**
  - Similar to ACID Properties, Consistency means that the state of the system before and after transactions should remain consistent.
- **Availability:**
  - This states that resources should always be available, there should be a non-error response.
- **Partition tolerance:**
  - Partition tolerance: Even when the network communication fails between the nodes in a cluster, the system should work well.

By the CAP theorem, all of these three properties cannot be achieved at the same time.

BOSSCODER ACADEMY

CAP THEOREM :



BOSSCODER ACADEMY

No-Sql Database

BASE PROPERTIES:

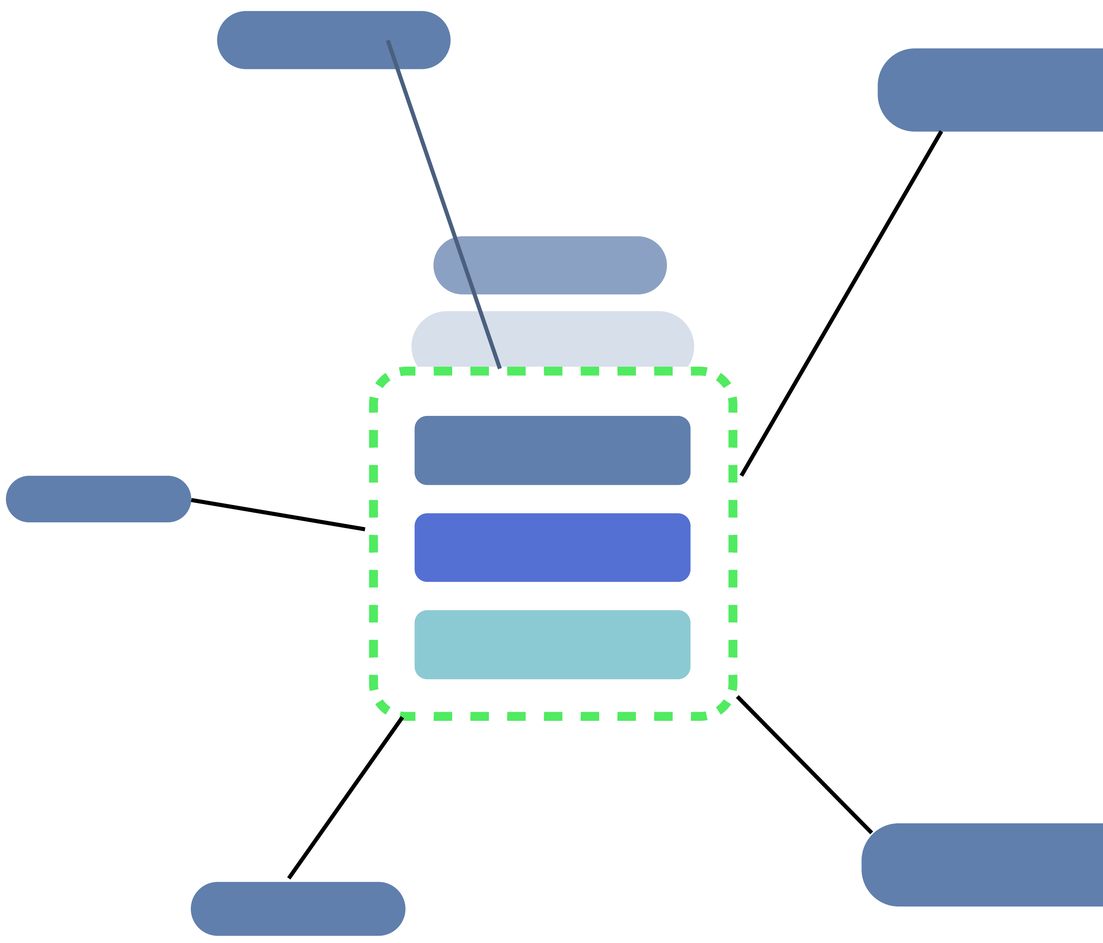
BASE stands for Basically Available, Soft state, Eventual consistency.

Basically Available:

Immediate consistency is not compulsory, but availability of data is basically guaranteed by replicating the database across different nodes of the database cluster.

Soft State:

Data values in the database may change over time. This is due to the fact that there is no immediate consistency, and the BASE model delegates the responsibility of consistency to developers.

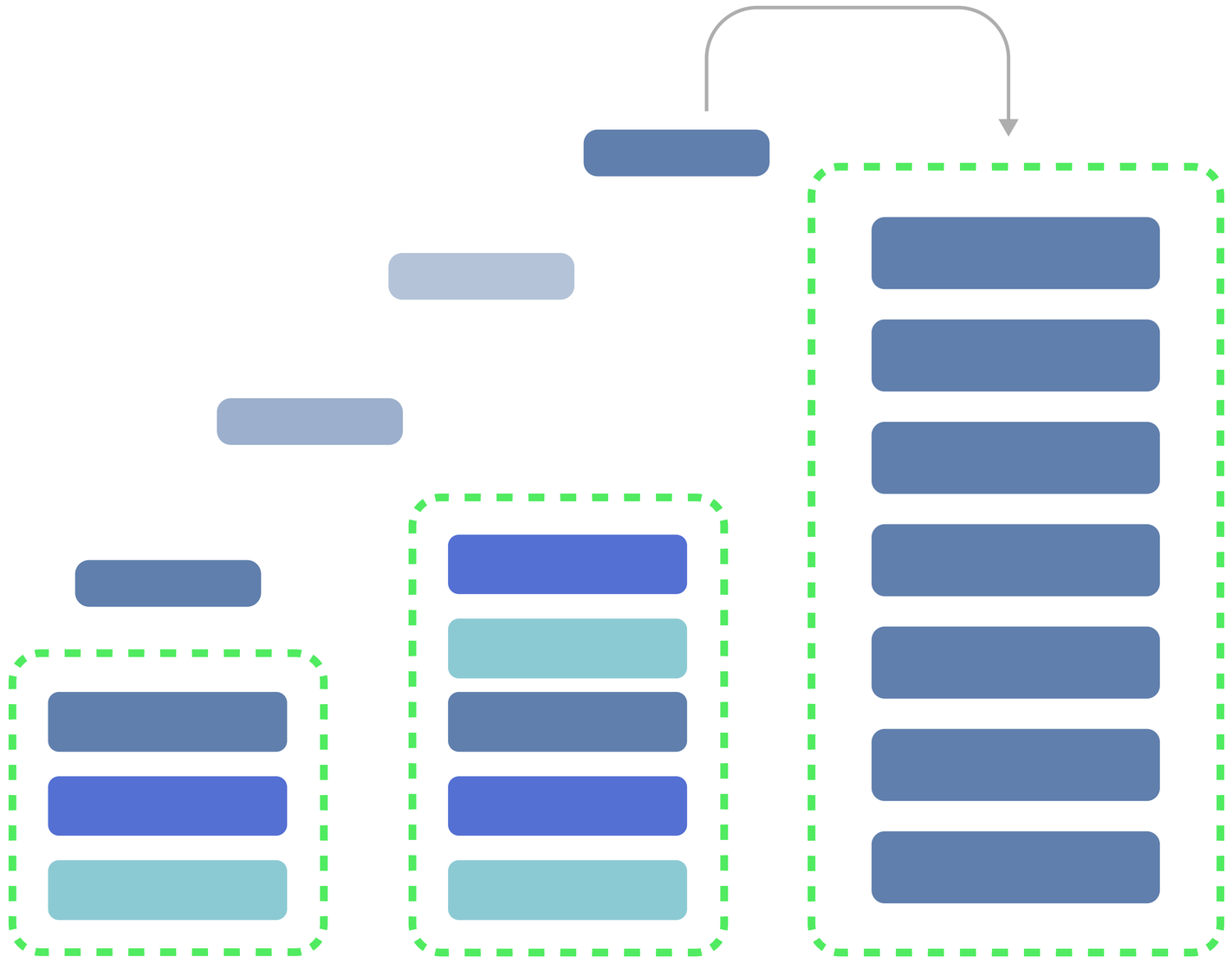


BOSSCODER ACADEMY

Eventually Consistent:

Even though the database does not guarantee immediate consistency, it doesn't mean that it is never achieved. Data read operations are still possible which may sometimes result in inconsistent information display.

NoSQL databases give up the Atomicity, Consistency and/or Durability requirements, and in return they improve scalability.



BOSSCODER ACADEMY

Sql & No-Sql

QUERYING LANGUAGES

SQL Databases :

Some relational database products support pure SQL.

Many include enhanced versions —such as SQL Server's Transact-SQL (T-SQL) for product specific features.

All SQL databases support the core ANSI/ISO language elements.

No- SQL Databases :

The language used by NoSQL depends on the type of NoSQL database and the type of operations required.

For example, MongoDB stores all documents in a JSON format. Queries are carried out using Javascript programming language.

BOSSCODER ACADEMY

SCHEMAS IN SQL & NO-SQL

SQL Databases :

SQL databases use predefined rigid schemas. They determine how tables are organised and data records are stored.

This results in a rigid structure that helps to optimise storage. This ensures data integrity but flexibility of schemas and models is limited.

NO-SQL Databases :

NoSQL databases use dynamic schemas.There are no predefined data structures for organising data.

This results in a high degree of flexibility, for example being able to add documents with different fields to the same database.

BOSSCODER ACADEMY

Sql & No-Sql

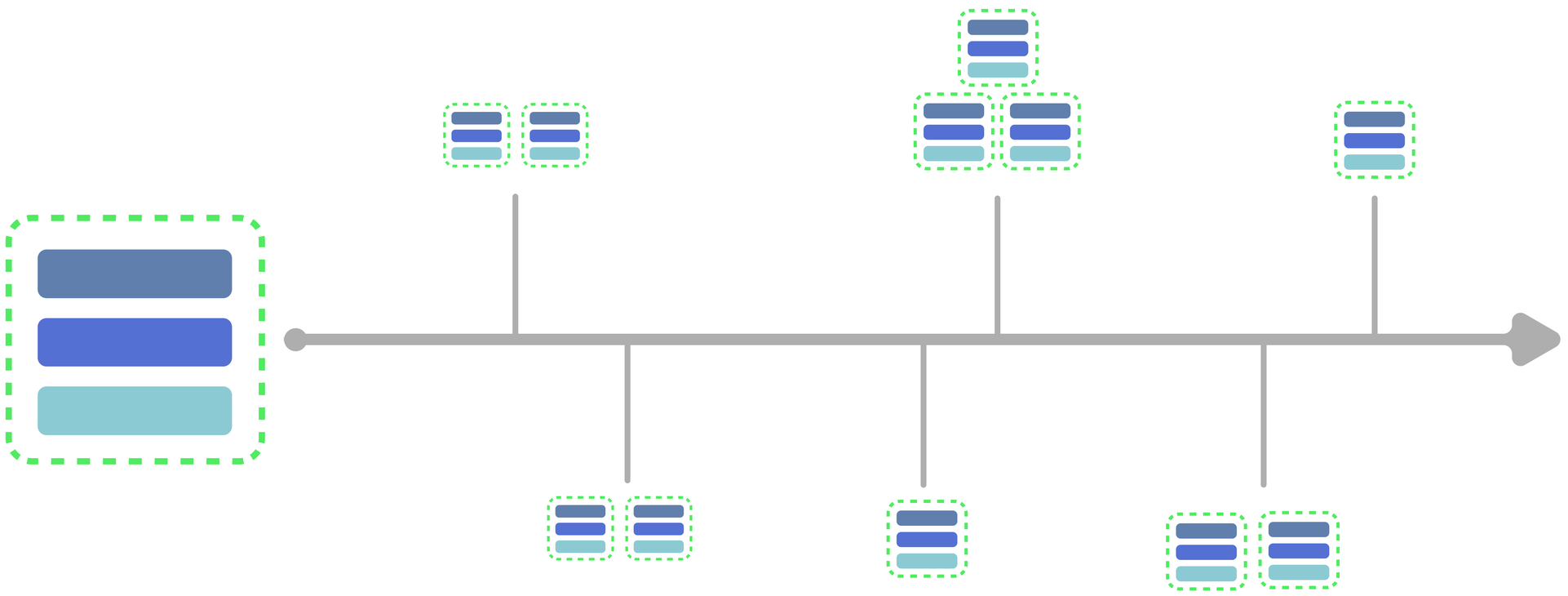
SCALABILITY IN DATABASES

Scalability in SQL Databases :

SQL databases mainly scale vertically.

This means they can be easily scaled up by adding resources such as faster CPUs or memory.

SQL databases however are not very efficient at scaling horizontally, over different systems and larger geographical distances.They are hence not suited for large, distributed data sets.



BOSSCODER ACADEMY

Scalability in NoSQL Databases :

NoSQL databases are able to scale horizontally very easily and efficiently across systems and geographical locations.

At the same time they also accommodate large stores of distributed data and easily support increased levels of traffic.

BOSSCODER ACADEMY

Sql & No-Sql

AGE & SUPPORT

SQL databases are built on technologies that have been supported by large developer communities over the years.

It is a relatively older database model as compared to NoSQL, and hence efficient documentation, developer activity is available.

NoSQL is not as old as SQL, however it is rapidly gaining support from big tech companies, for efficient querying.

However different companies have been steadily incorporating features of both types of databases into their products to make them more universal.

For Example

For example, MongoDB now supports multi-document ACID transactions, and MySQL now includes a native JSON data type for storing and validating JSON documents.

BOSSCODER ACADEMY

When to use

SQL VS NO-SQL ?

When to use SQL:

- For highly structured data or relational data without the complexity of multiple data stores
- Transaction based systems are used like financial transfers
- When data security and integrity is required to be of high level
- When complex queries are routinely executed
- When scale out capabilities offered by NoSQL is not required

BOSSCODER ACADEMY

When to use NO-SQL:

- When large amounts of unstructured or semi-structured data are used
- When dynamic schema is desired or flexibility of data model is required
- When high level of integrity of data offered by SQL database is not required
- When database has to be scaled horizontally, over various locations locationally
- To avoid overhead of structured database model

BOSSCODER ACADEMY

WHY BOSSCODER

200+ alumni placed at **Top product- based companies**

136% (Avg) hike for **working professionals**

22 LPA (Avg) package for **Bosscoder alumni**

Lavanya Meta

The syllabus is most up-to-date and the list of problems provided covers all important topics.

Rahul Google

Course is very well structured and streamlined to crack any MAANG company