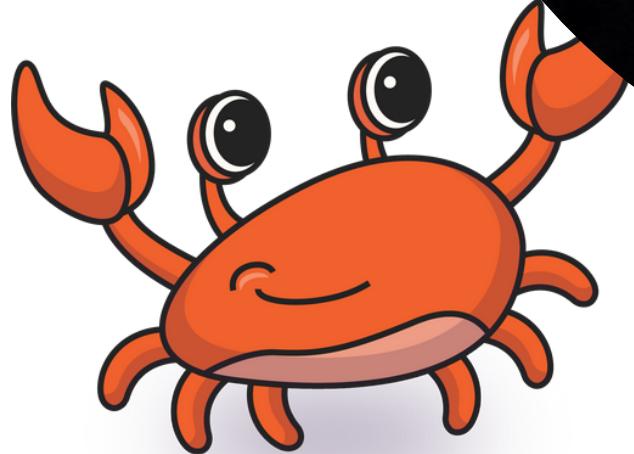


# NAVIGATING OPEN DATABASES



by @laysauchoa

# SELF.INTRO()

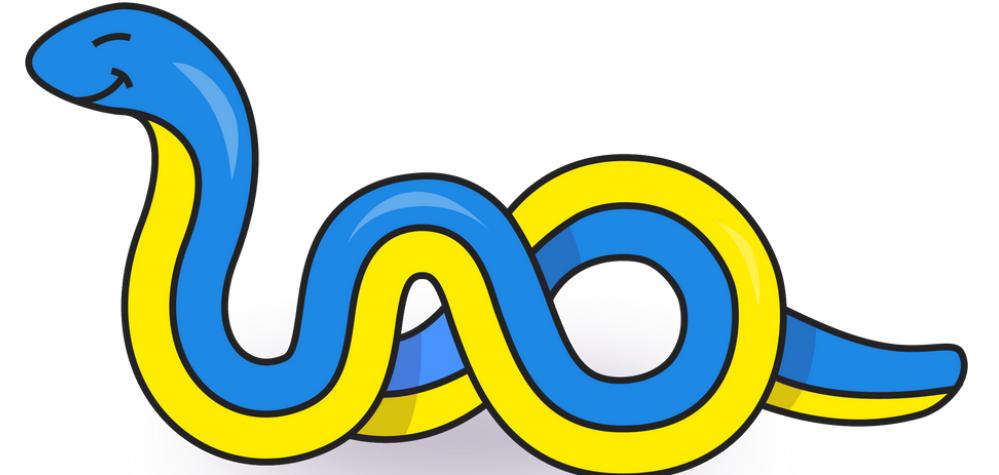


AIVEN

OSS DATABASES

*pyladies* 

by @laysauchoa

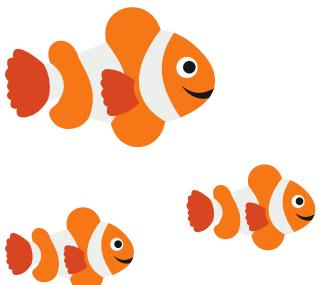
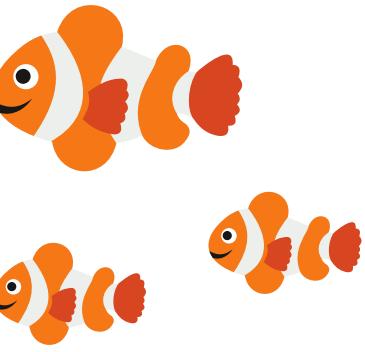


# AGENDA

HISTORY

PARADIGMS

FINAL THOUGHTS



# HISTORY



Charles Bachman

Navigational DBMS

IBM

1960

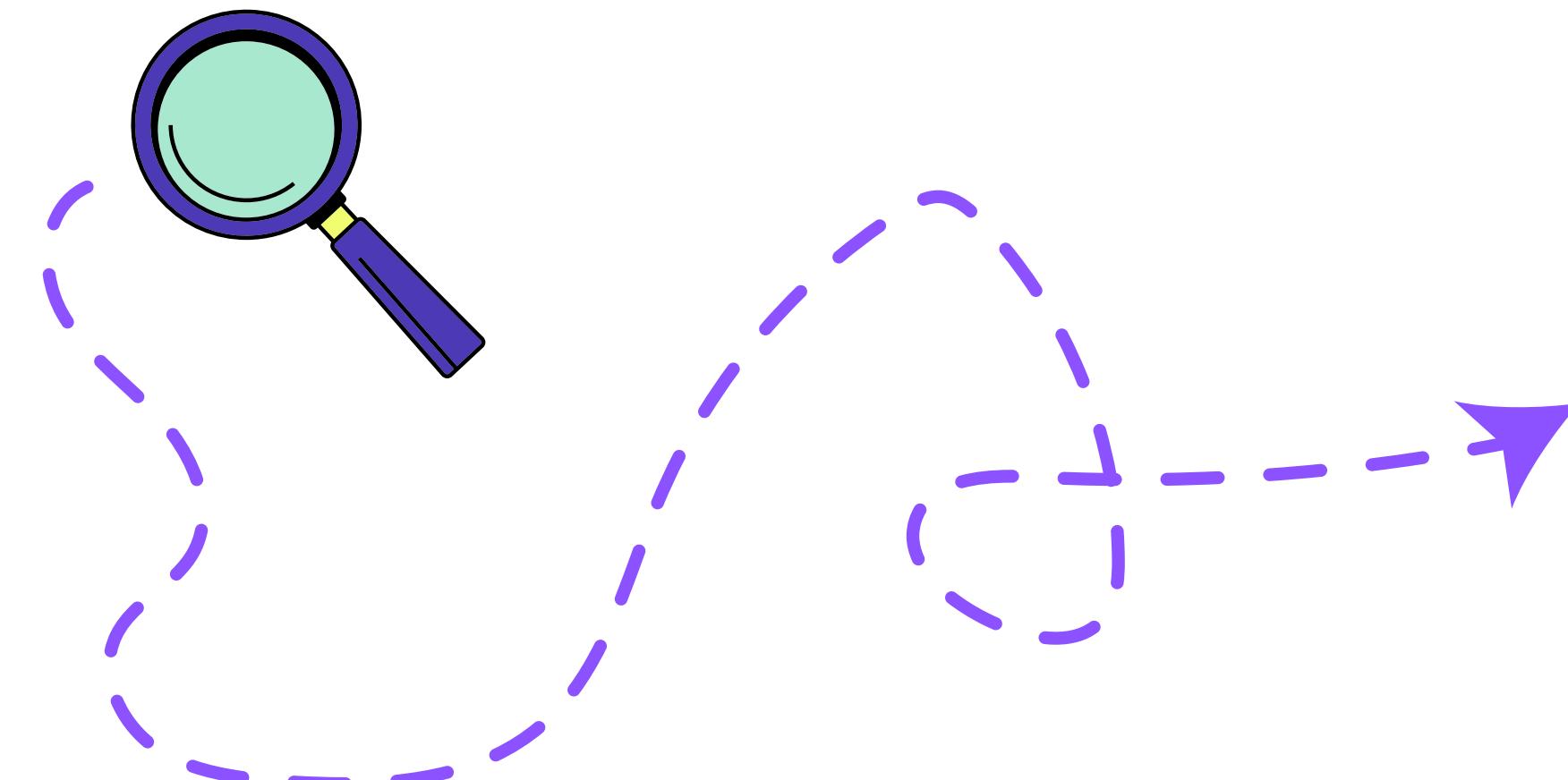


Charles Bachman

Navigational DBMS

IBM

1960

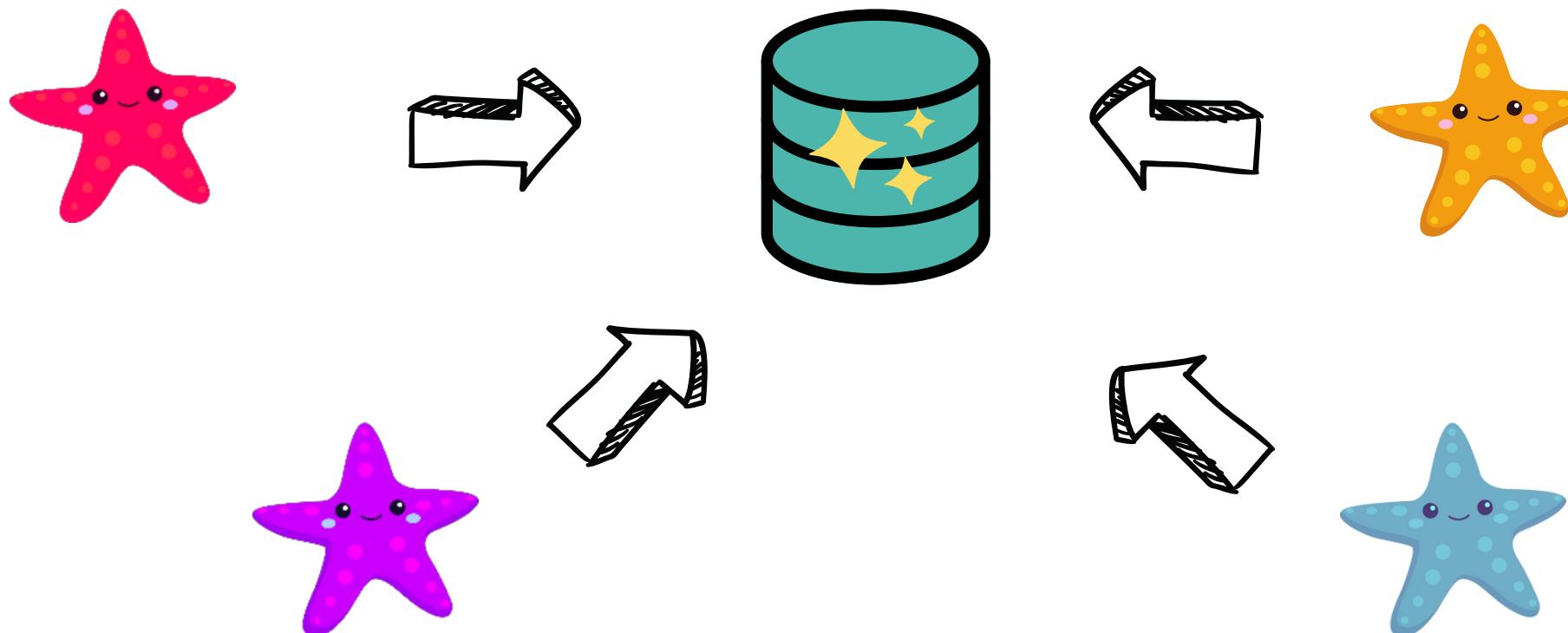


# Charles Bachman

# Navigational DBMS

1960

## Databases



Charles Bachman

Navigational DBMS

1960

Database Task Group

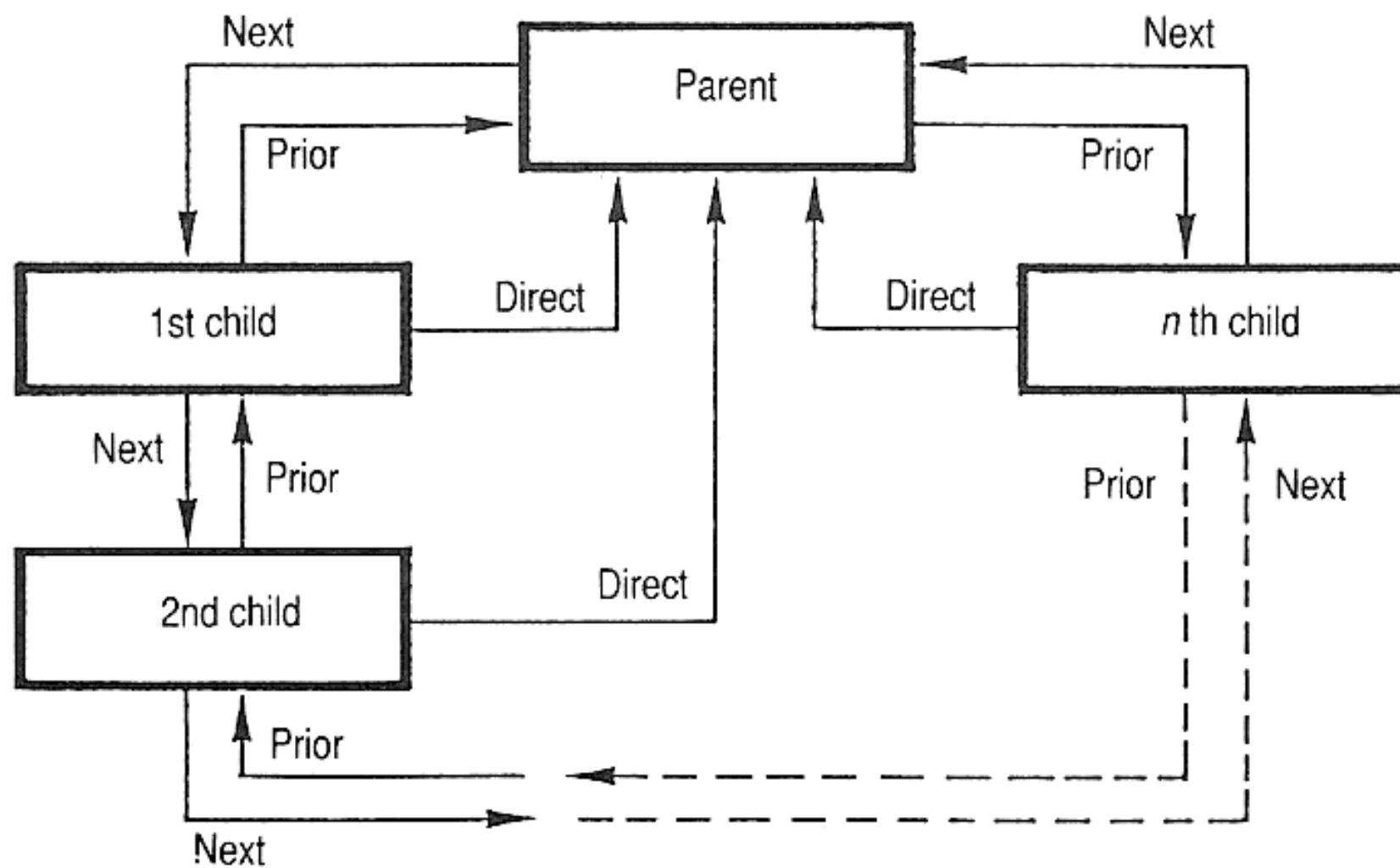
CODASYL



# Charles Bachman

# navigational DBMS

1960



# CODASYL



# Edgar Codd

1960 > 1970



# Relational databases

## *Information Retrieval*

### A Relational Model of Data for Large Shared Data Banks

E. F. CODD

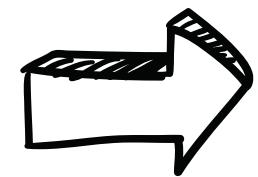
*IBM Research Laboratory, San Jose, California*

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed

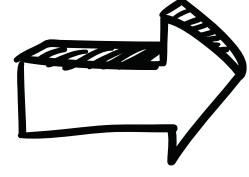
# Edgar Codd

1960 → 1970

data



data



data

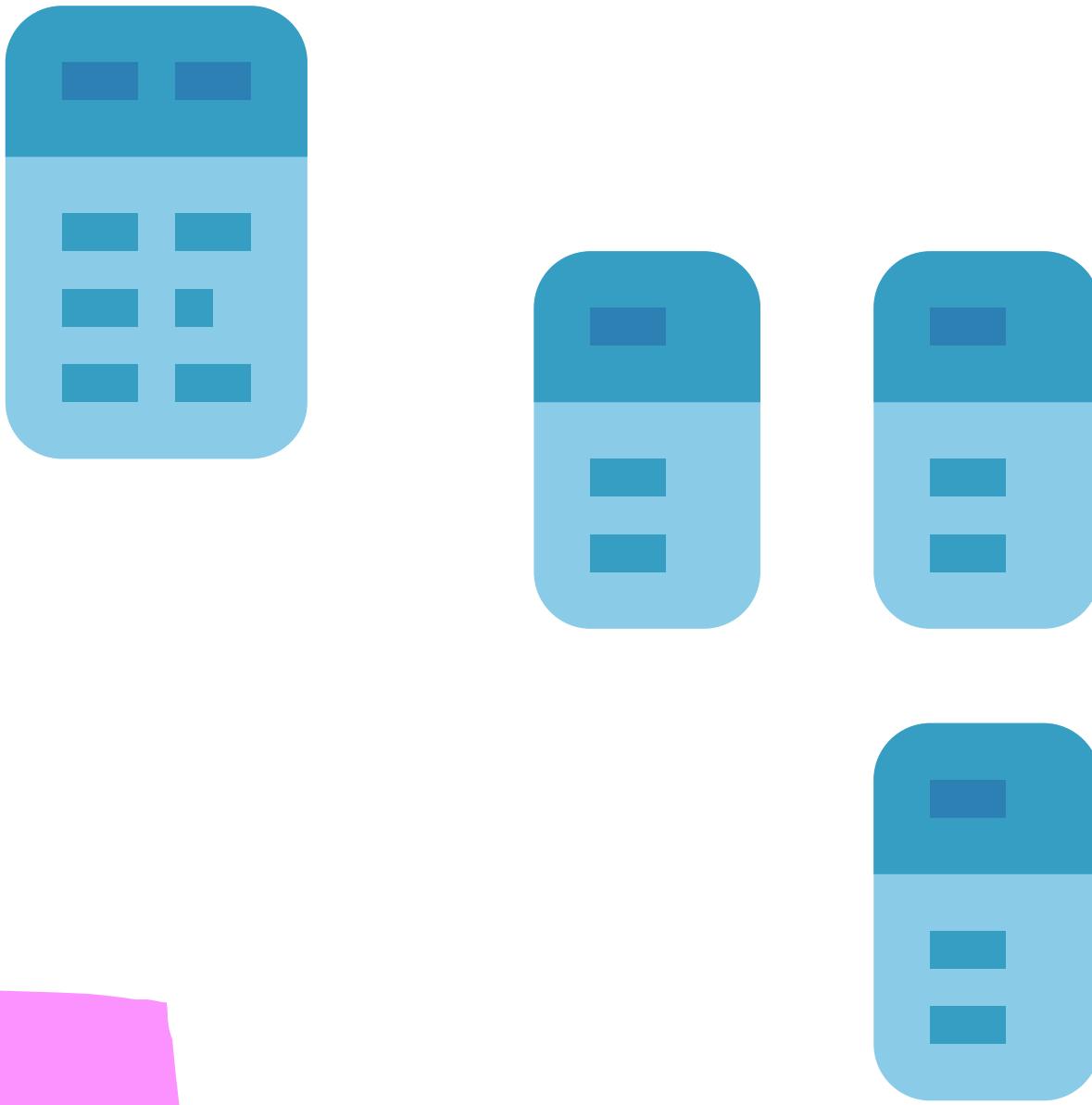


# Edgar Codd

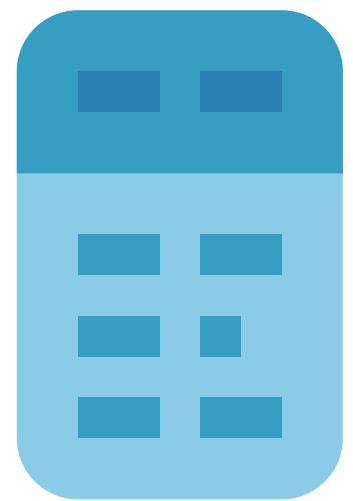
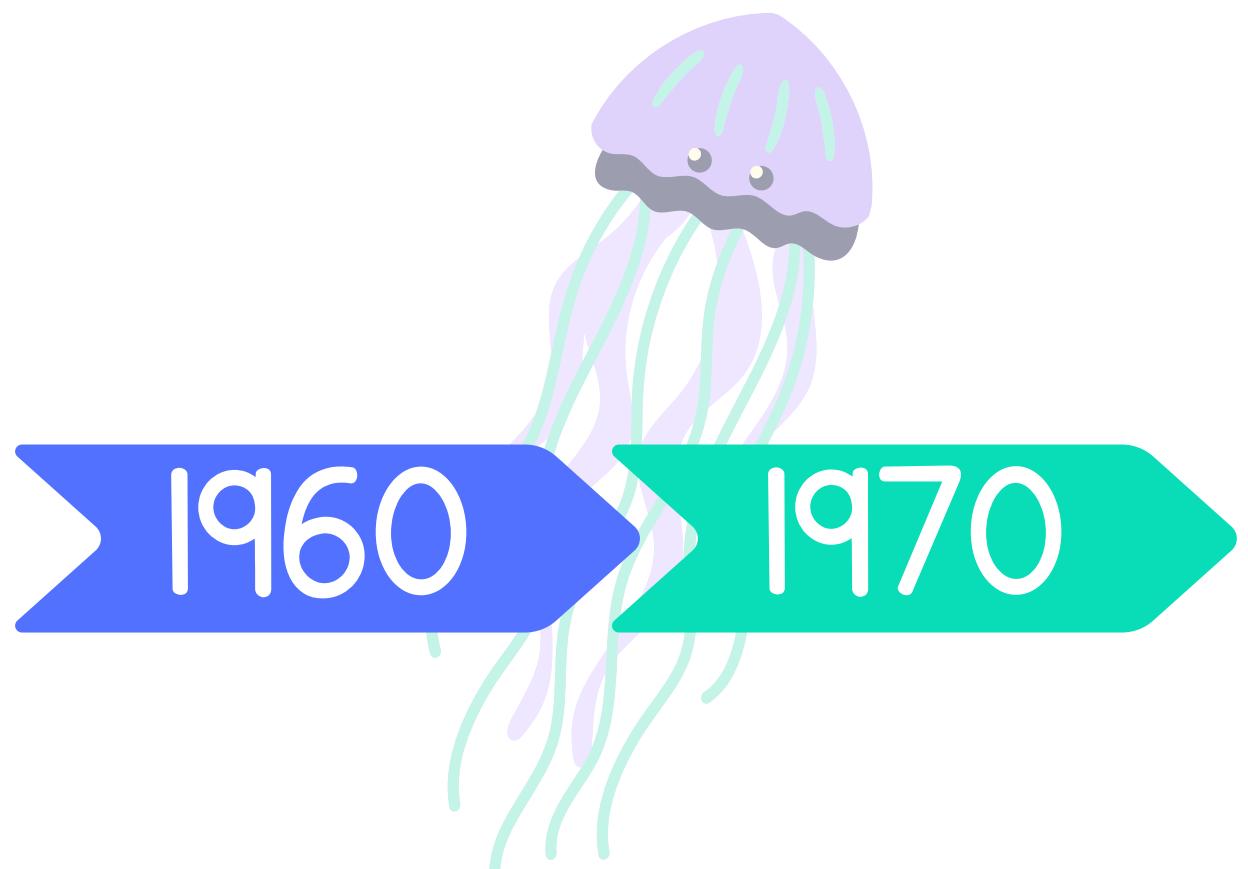
1960 → 1970



tables



# Edgar Codd



entity

tables

ID	name	age
112	bea	34
113	pablo	36
114	caio	21

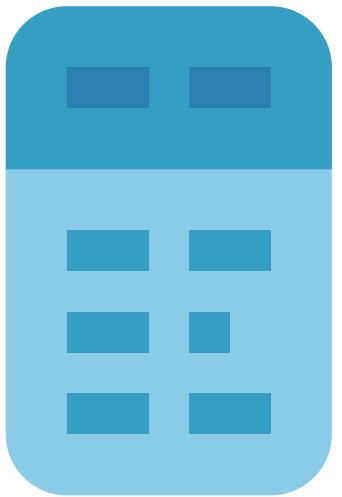
People



Edgar Codd

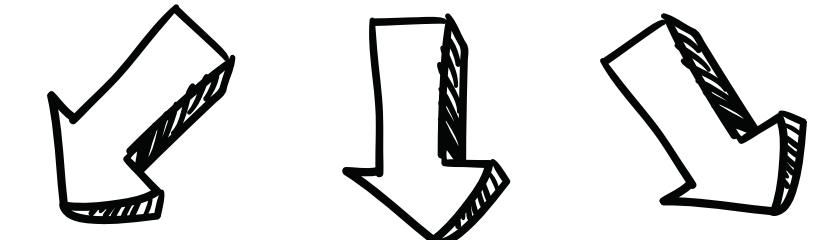
1960 → 1970

tables



entity

attributes



ID	name	age
112	bea	34
113	pablo	36
114	caio	21

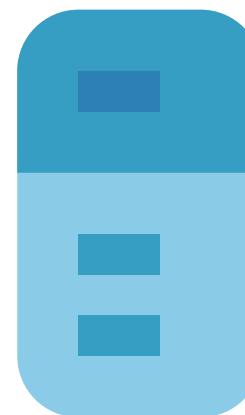
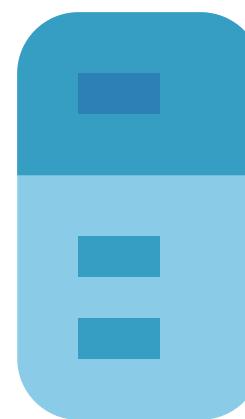
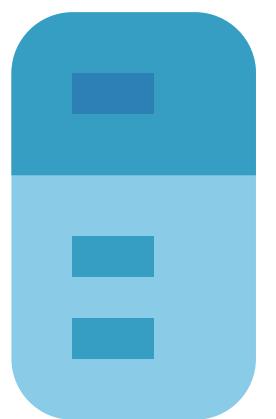
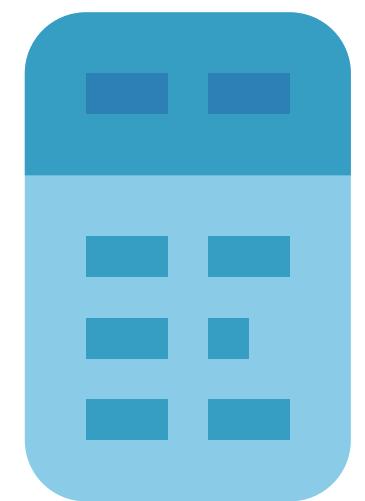
People

Edgar Codd



primary keys

1960 → 1970

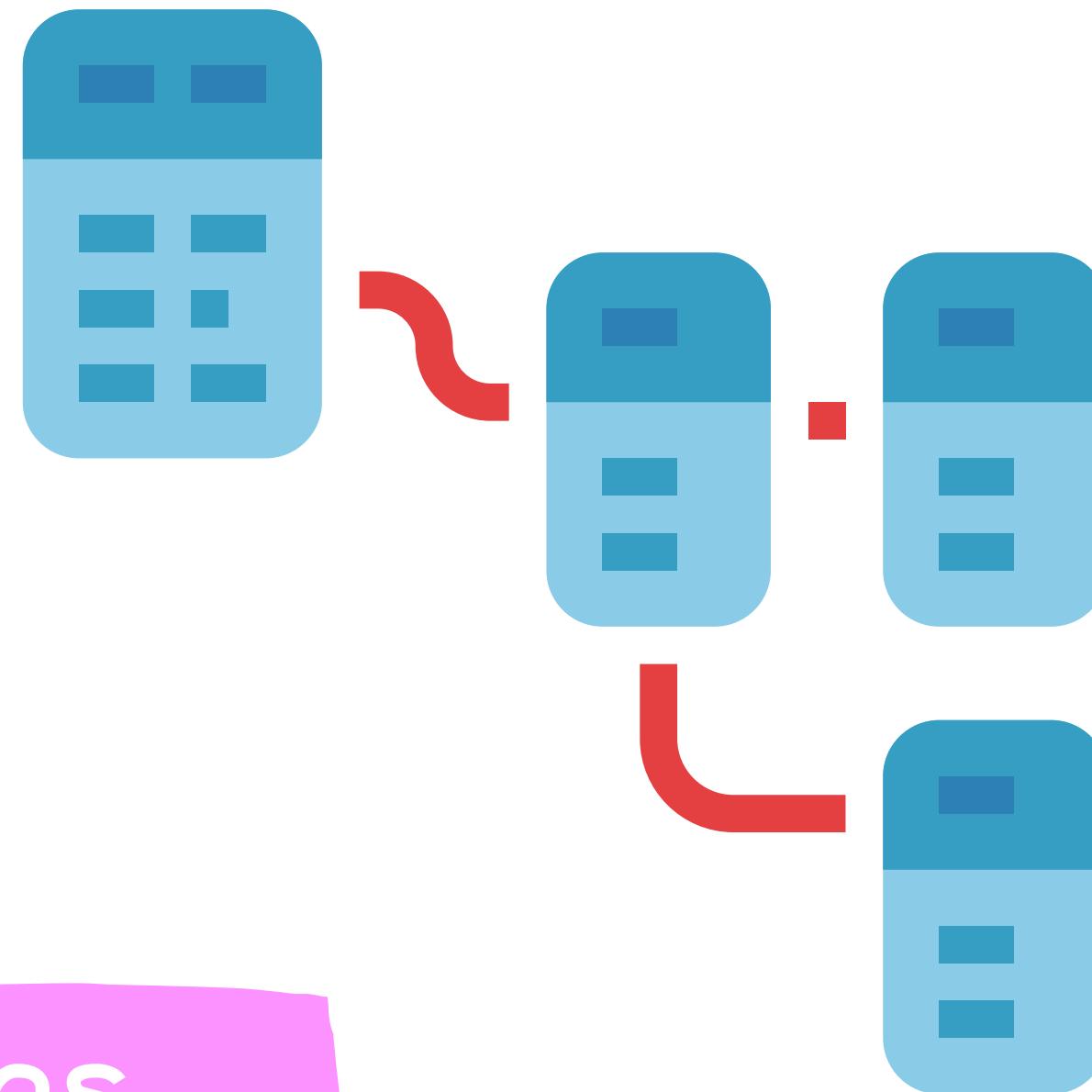


tables

Edgar Codd

primary keys

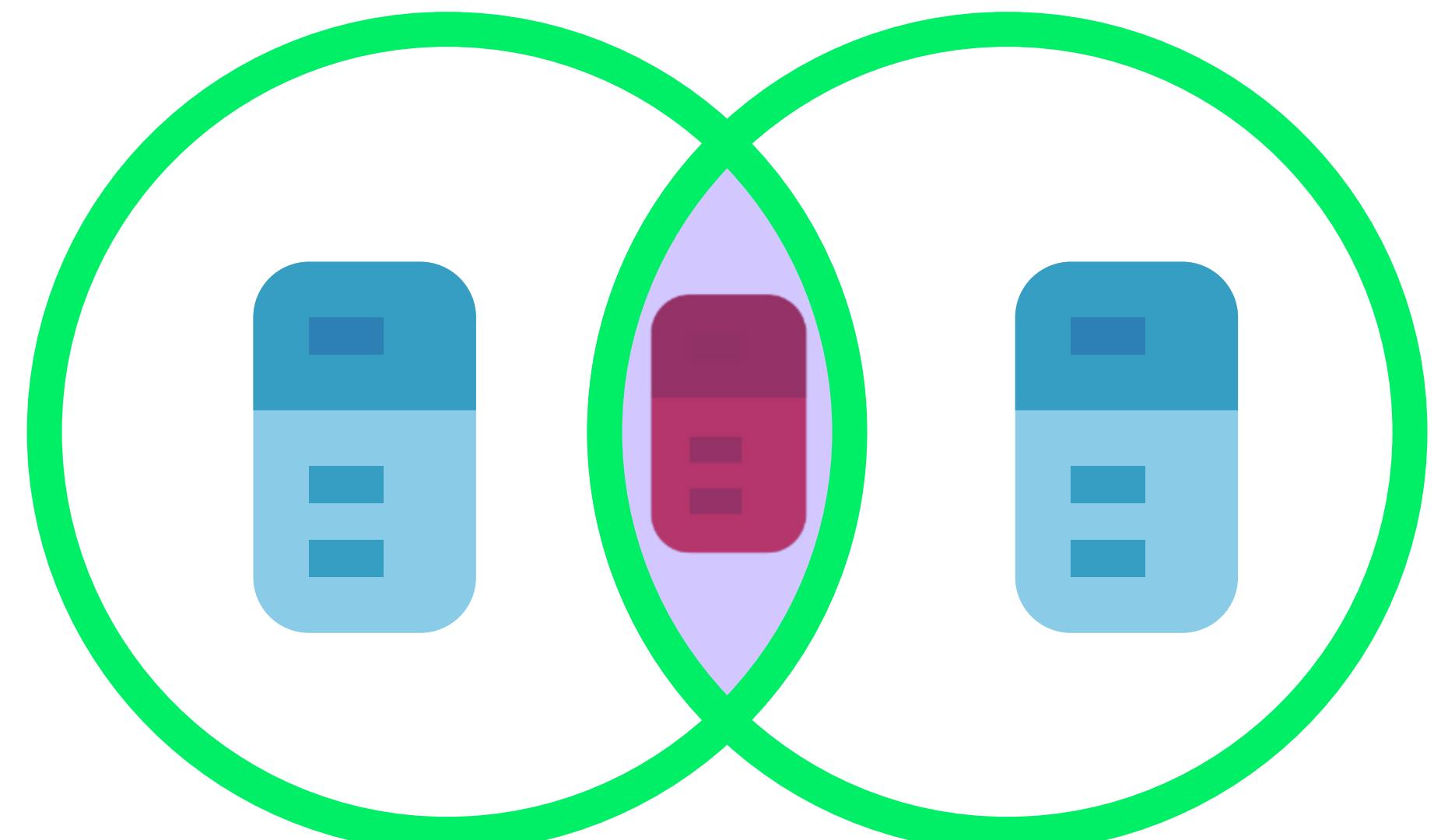
1960 → 1970



Edgar Codd

primary keys

1960 → 1970



# Edgar Codd

## Views

1960

1970



Wong and

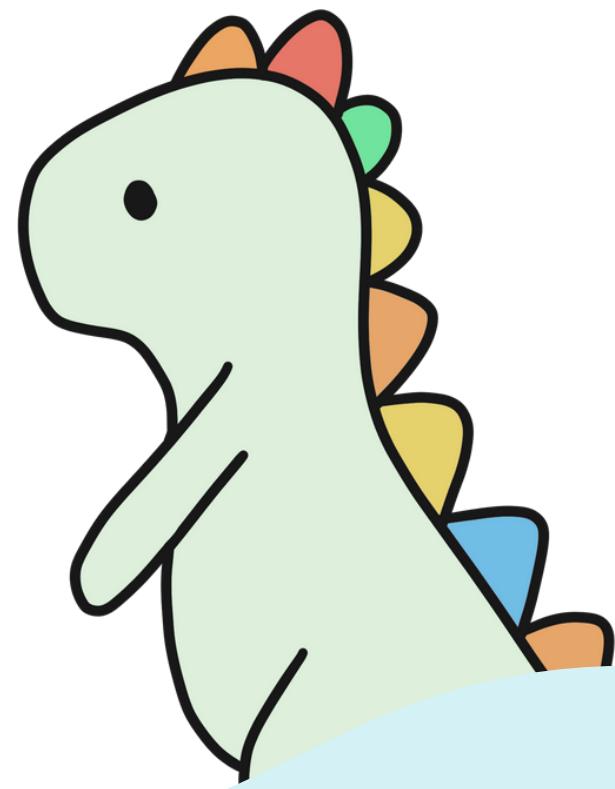
Stonebraker

1960

1970

INGRES

QUEL



# QUEL

```
create student(name = c10,  
age = i4, sex = c1, state =  
c2)
```

```
retrieve (s.all)
```

```
delete s where  
s.name="philip"
```

# SQL

```
create table student(name  
char(10), age int, sex char(1),  
state char(2));
```

```
select * from student;
```

```
delete from student where  
name='philip';
```

Stonebraker

Postgres

1970

1980



Postgres

Stonebraker

PostgreSQL

1970

1980



PostgreSQL

Stonebraker

PostgreSQL

Postgres

PostgreSQL



1970 → 1980

# relational databases

SQL

1960 → 1970 → 1980



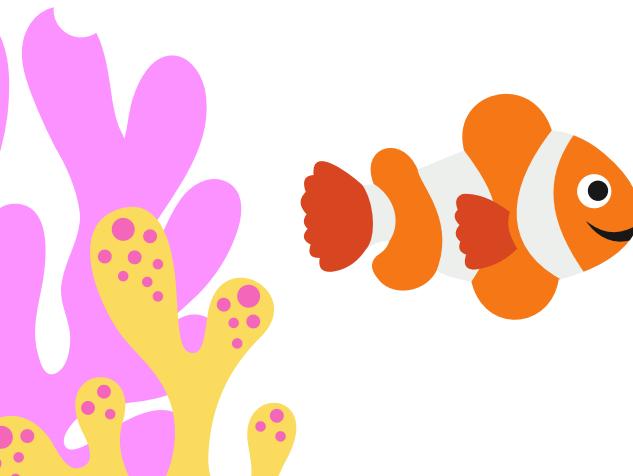
object db

1960

1970

1980

1990



object db

relational db

1960

1970

1980

1990

Mismatches



Johan Oskarsson

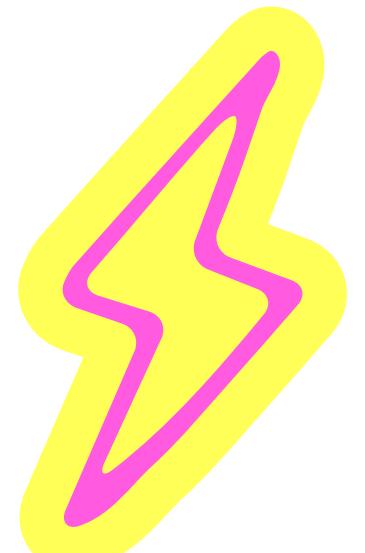
Voldemort

Cassandra

Dynomite



2000 → 2010 → 2020



# Johan Oskarsson

No schema

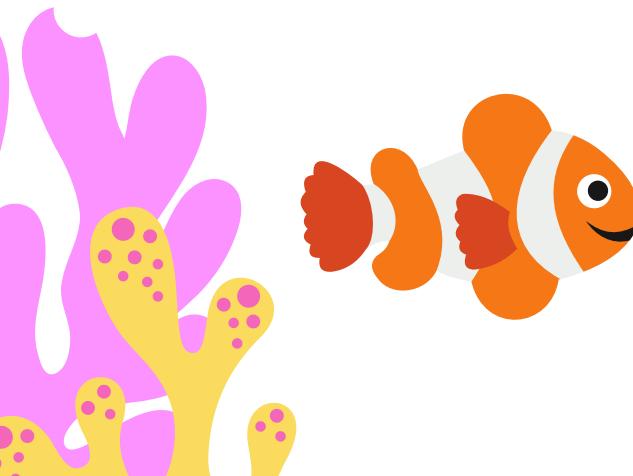
Distributed

No relational

2000

2010

2020



# Johan Oskarsson

No schema

Distributed

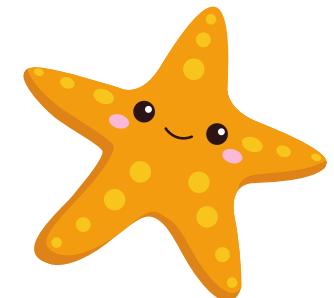
No relational

2000

2010

2020

NOSQL



NOSQL

No SQL

Not Only SQL

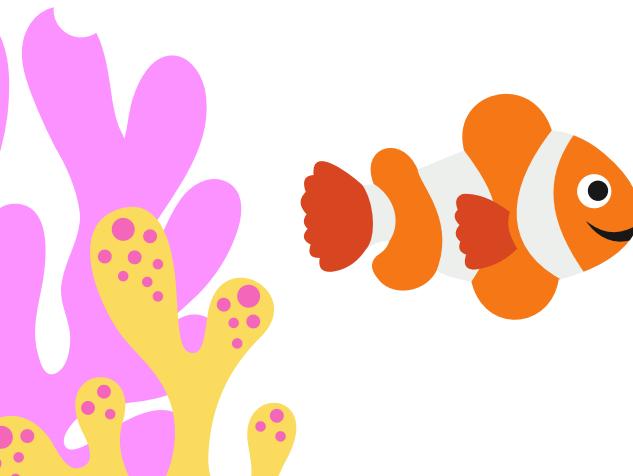


SQL

2000

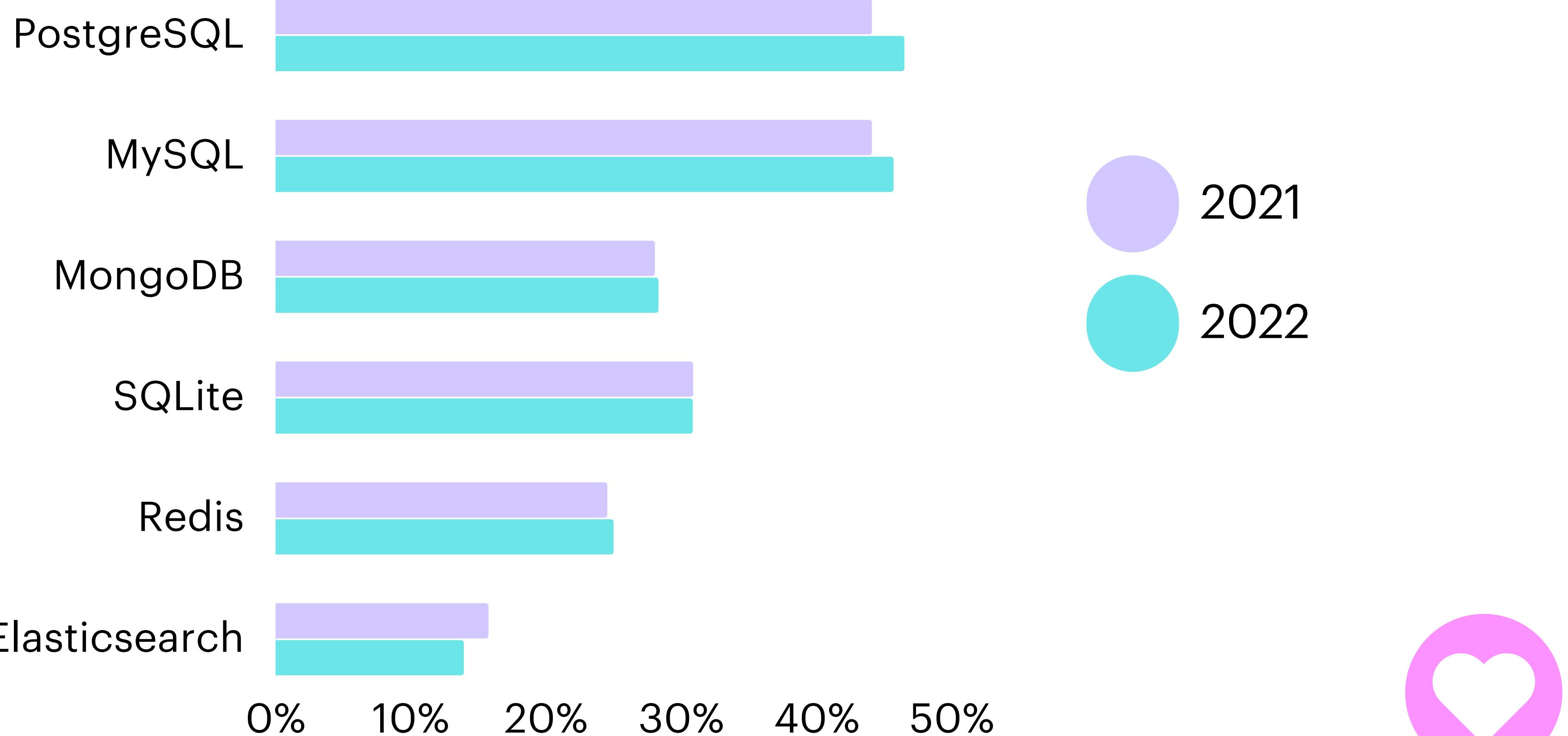
2010

2020



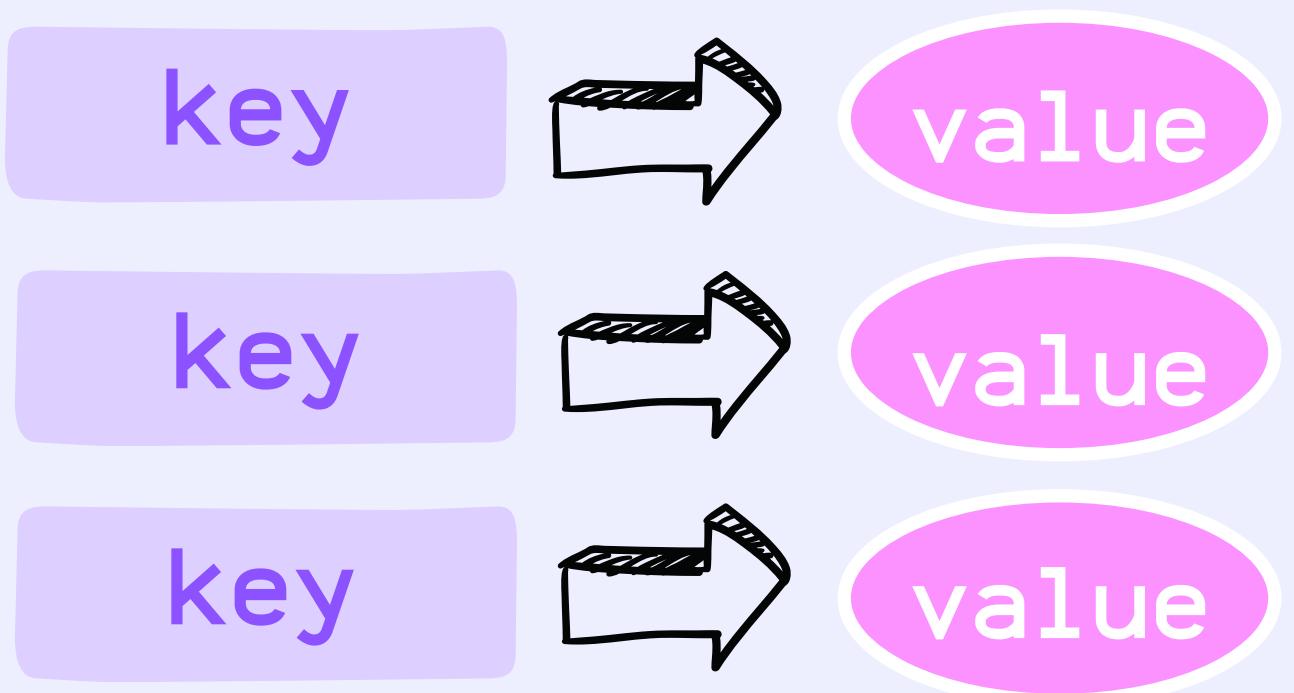
# DATABASES



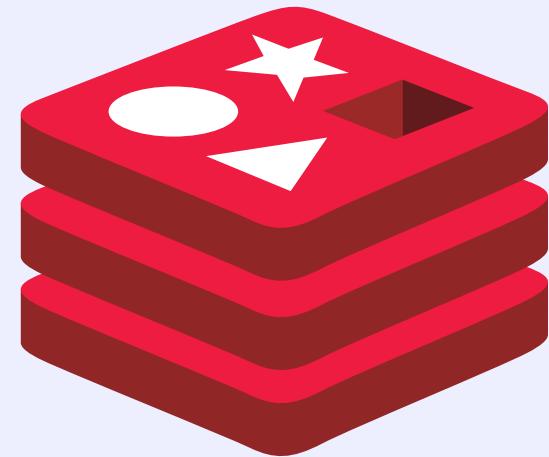


<https://survey.stackoverflow.co/>

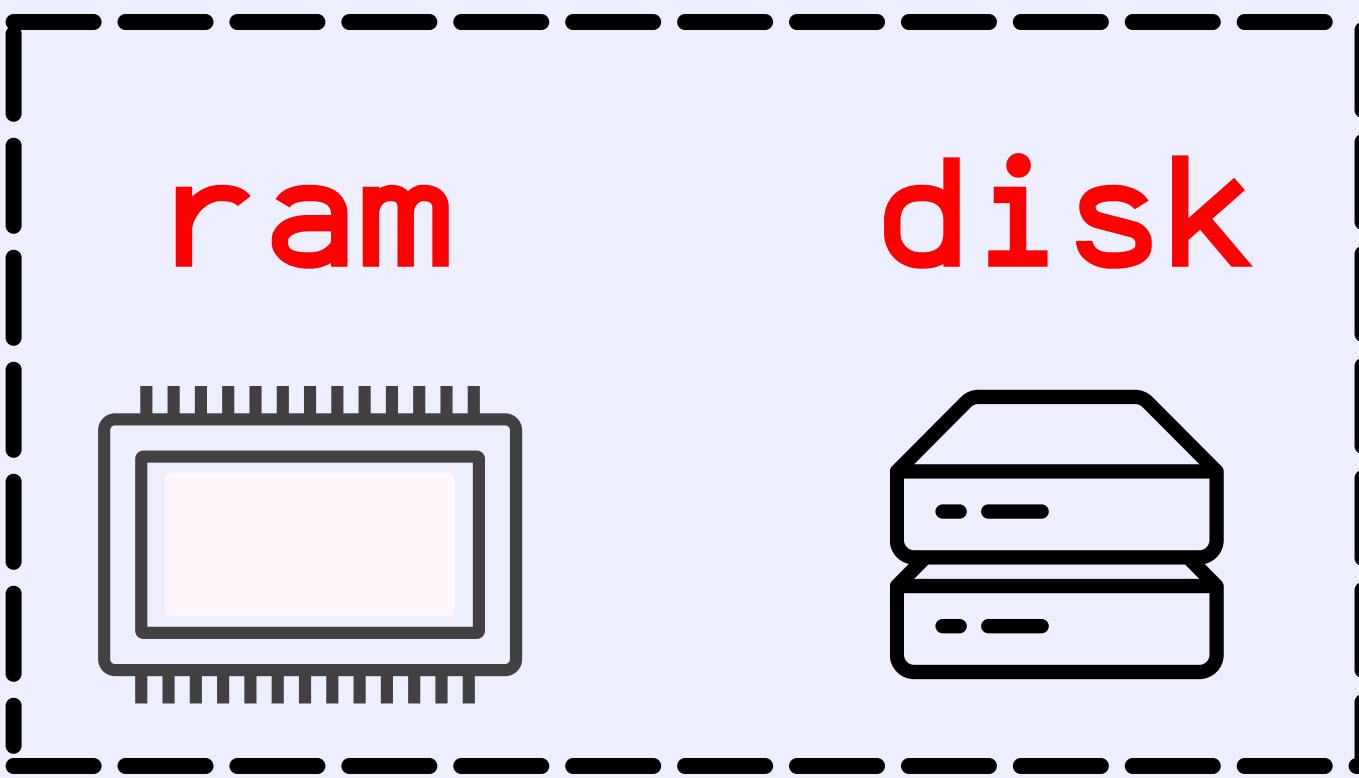
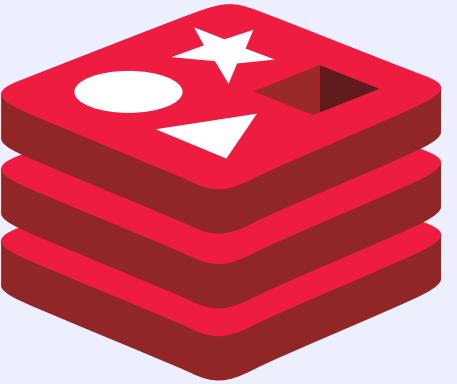
# KEY VALUE



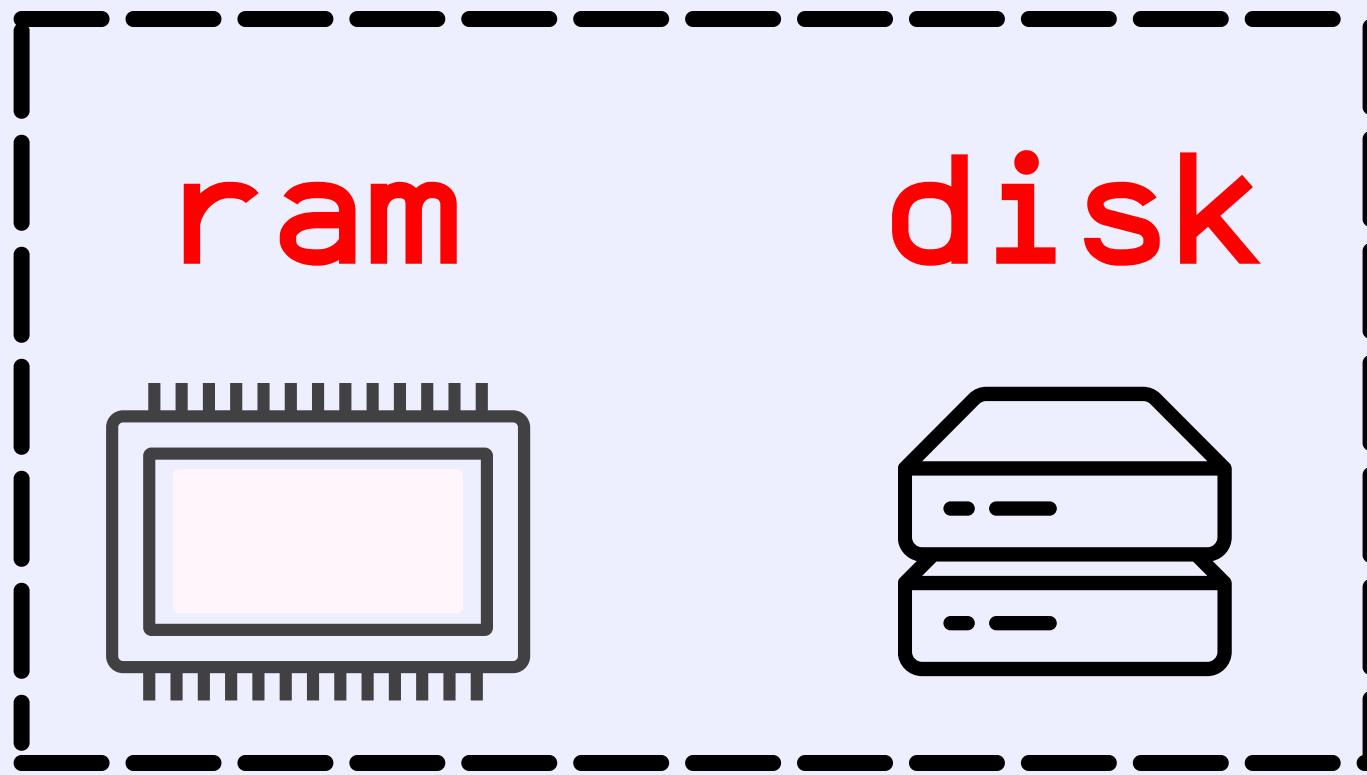
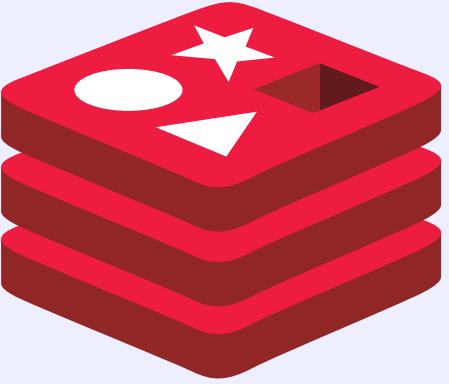
key  
name : starfish  
value



# IN-MEMORY



# IN-MEMORY

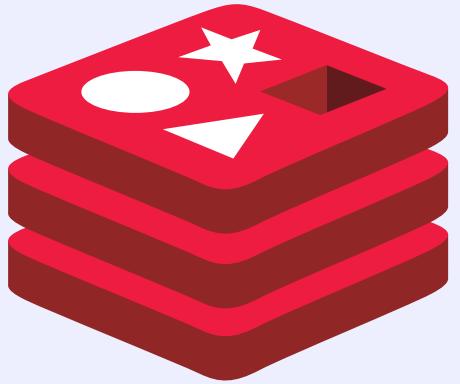


# KEY VALUE

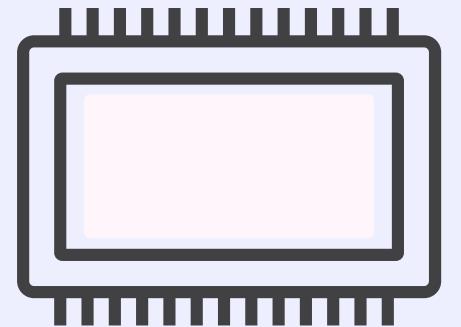
CACHING

PUB/SUB

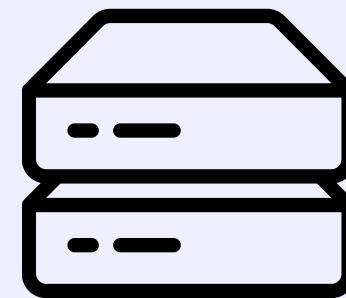
LEADERBOARDS



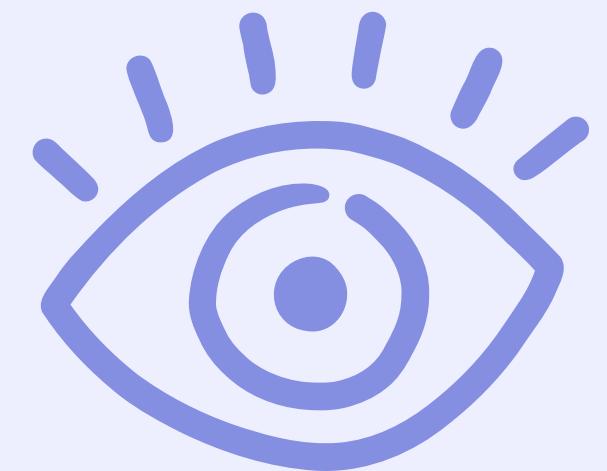
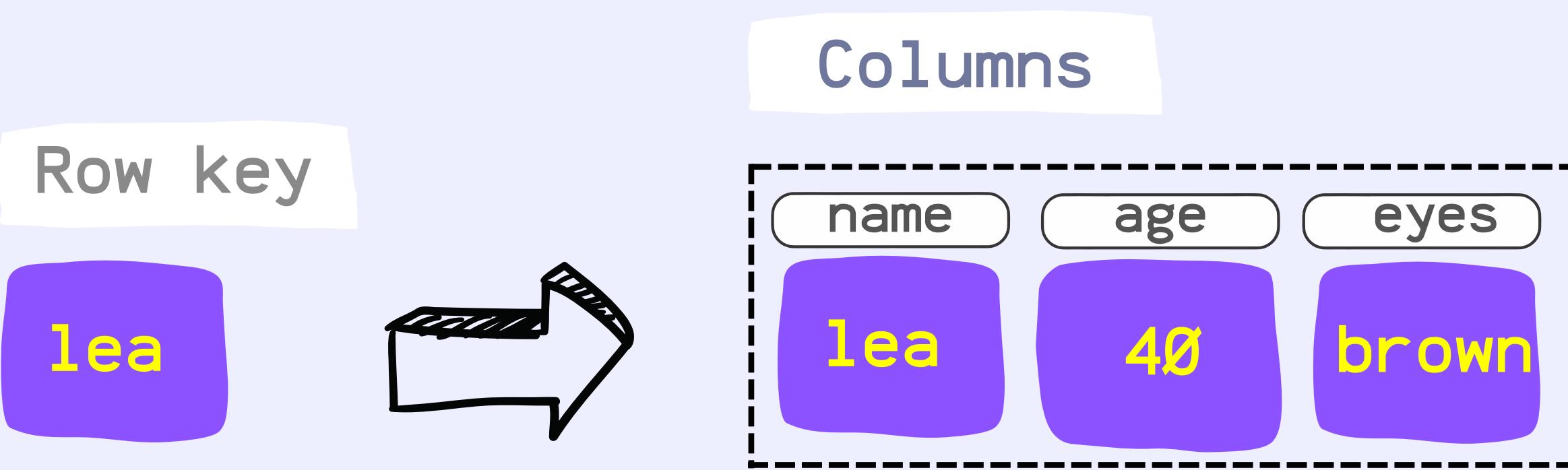
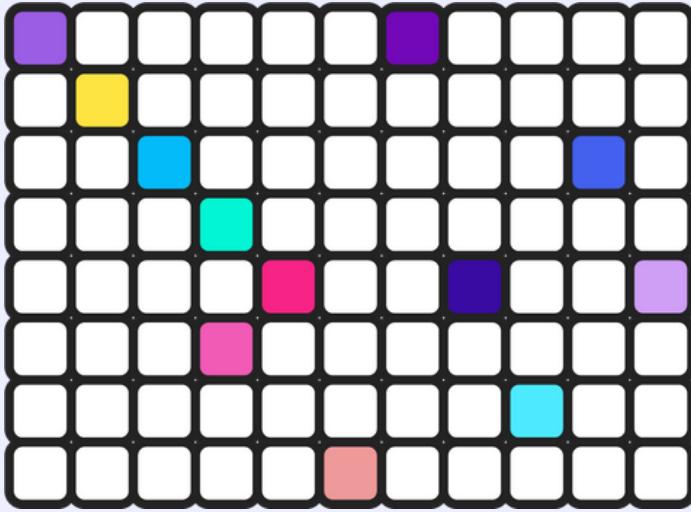
ram



disk



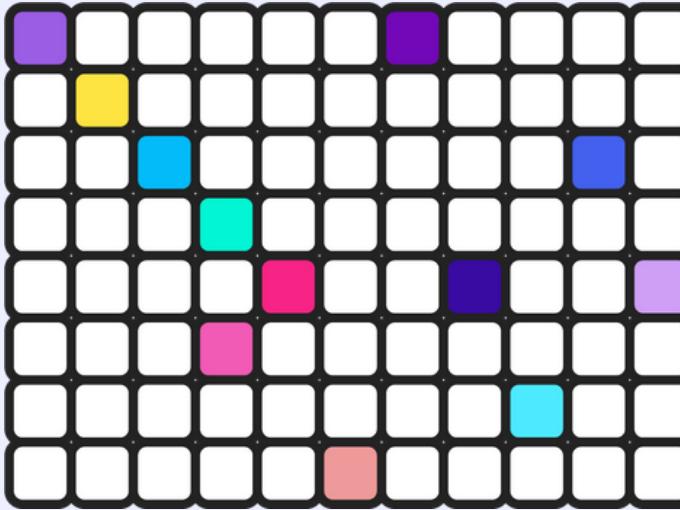
# WIDE COLUMN



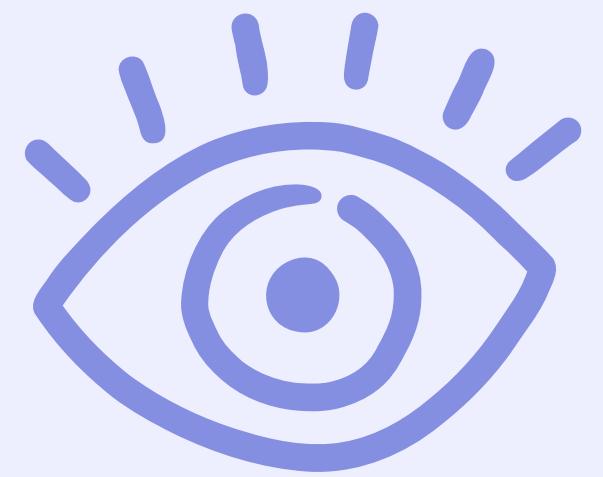
Cassandra



# WIDE COLUMN



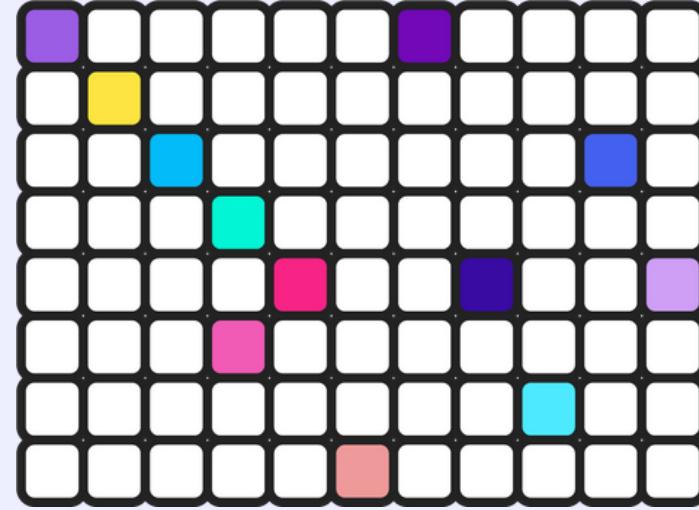
```
SELECT * FROM playlists  
WHERE id = 62c36092-82a1  
  
ORDER BY song_order DESC  
LIMIT 50;
```



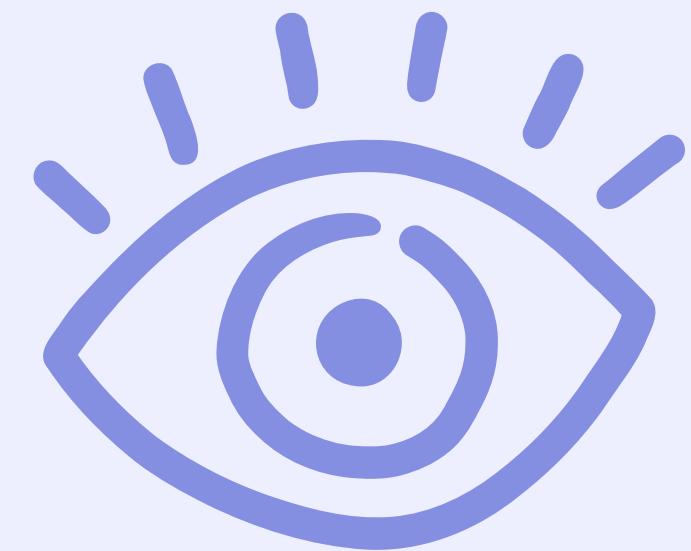
Cassandra

CQL

# WIDE COLUMN



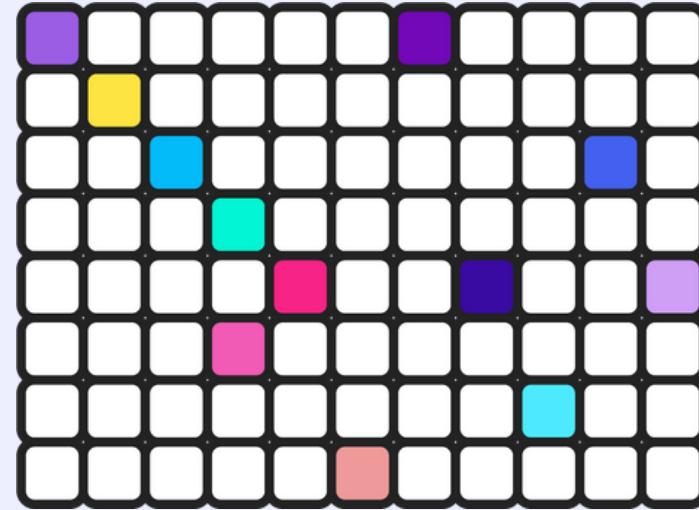
Schema-less



Cassandra

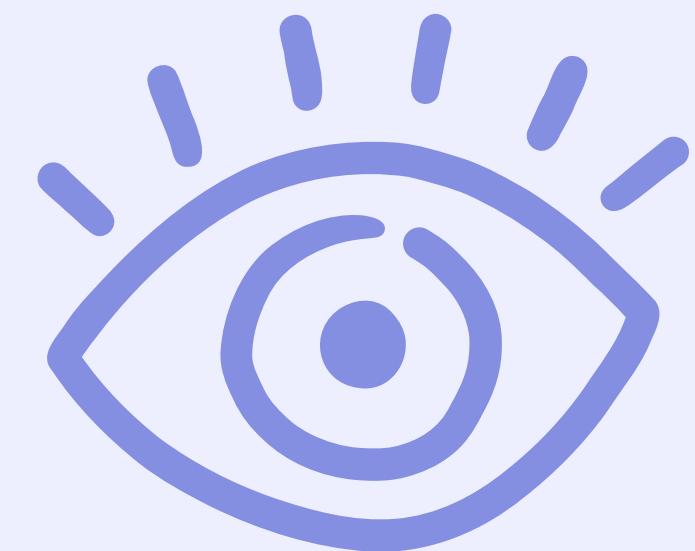


# WIDE COLUMN



Schema-less

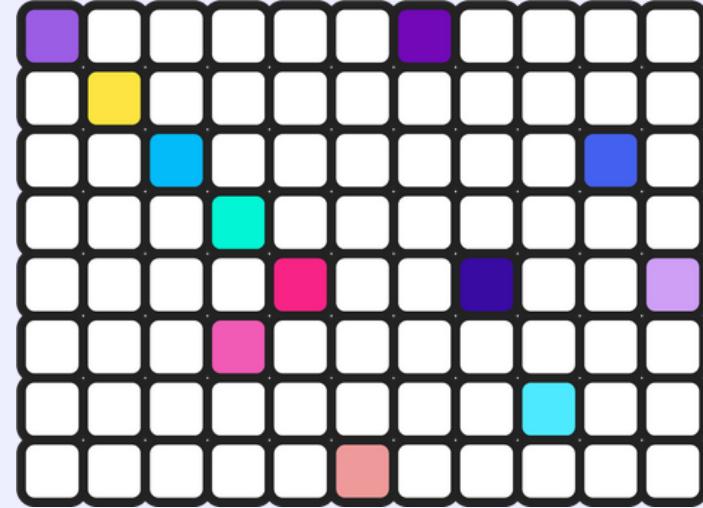
No joins



Cassandra



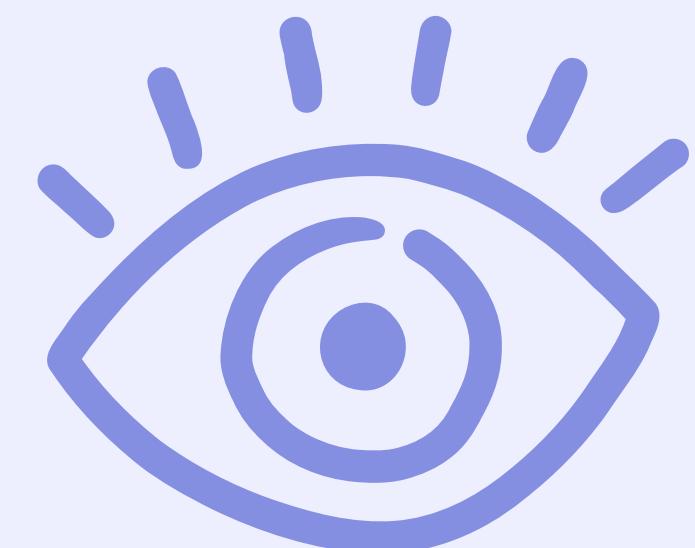
# WIDE COLUMN



Schema-less

No joins

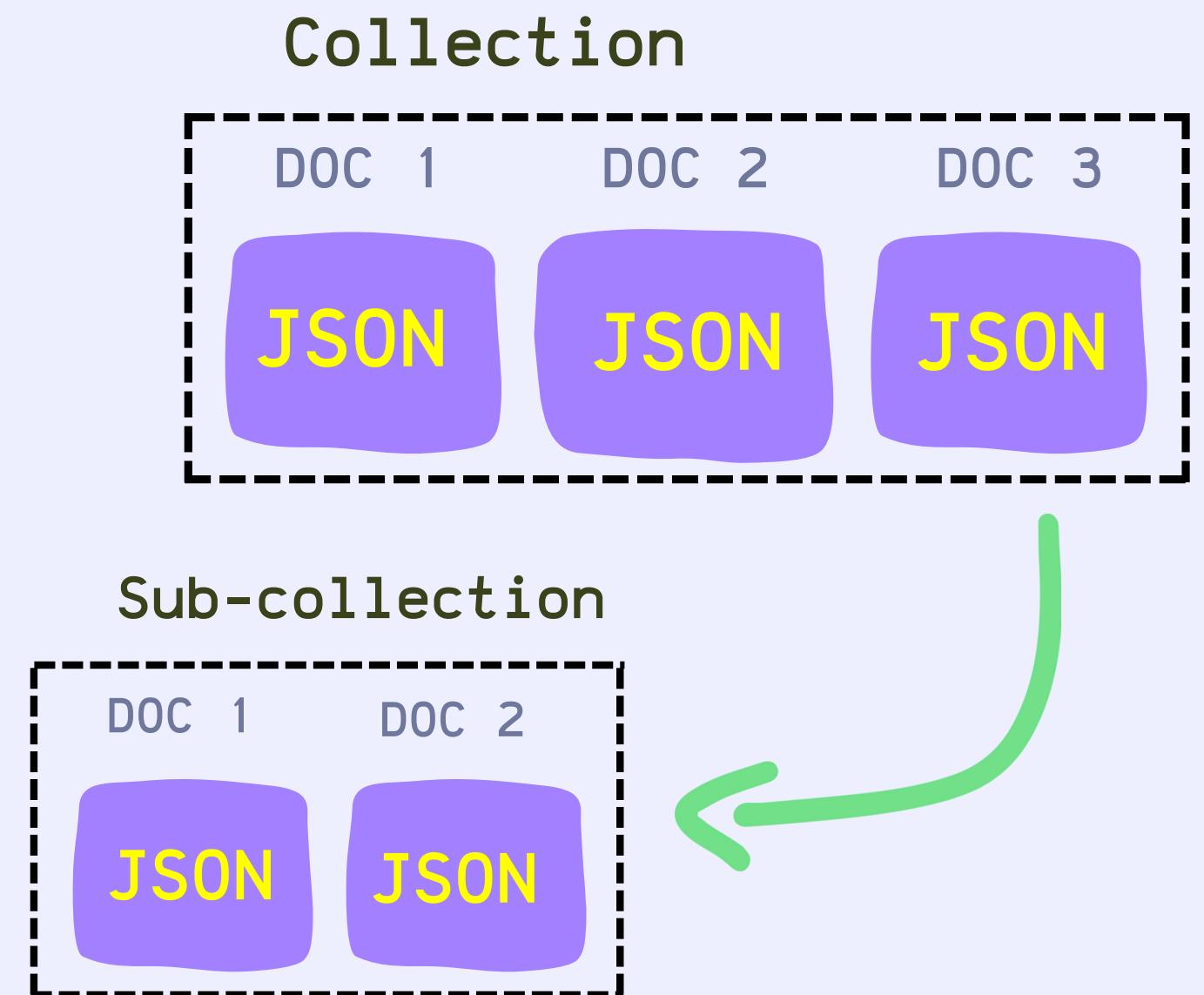
Scalability



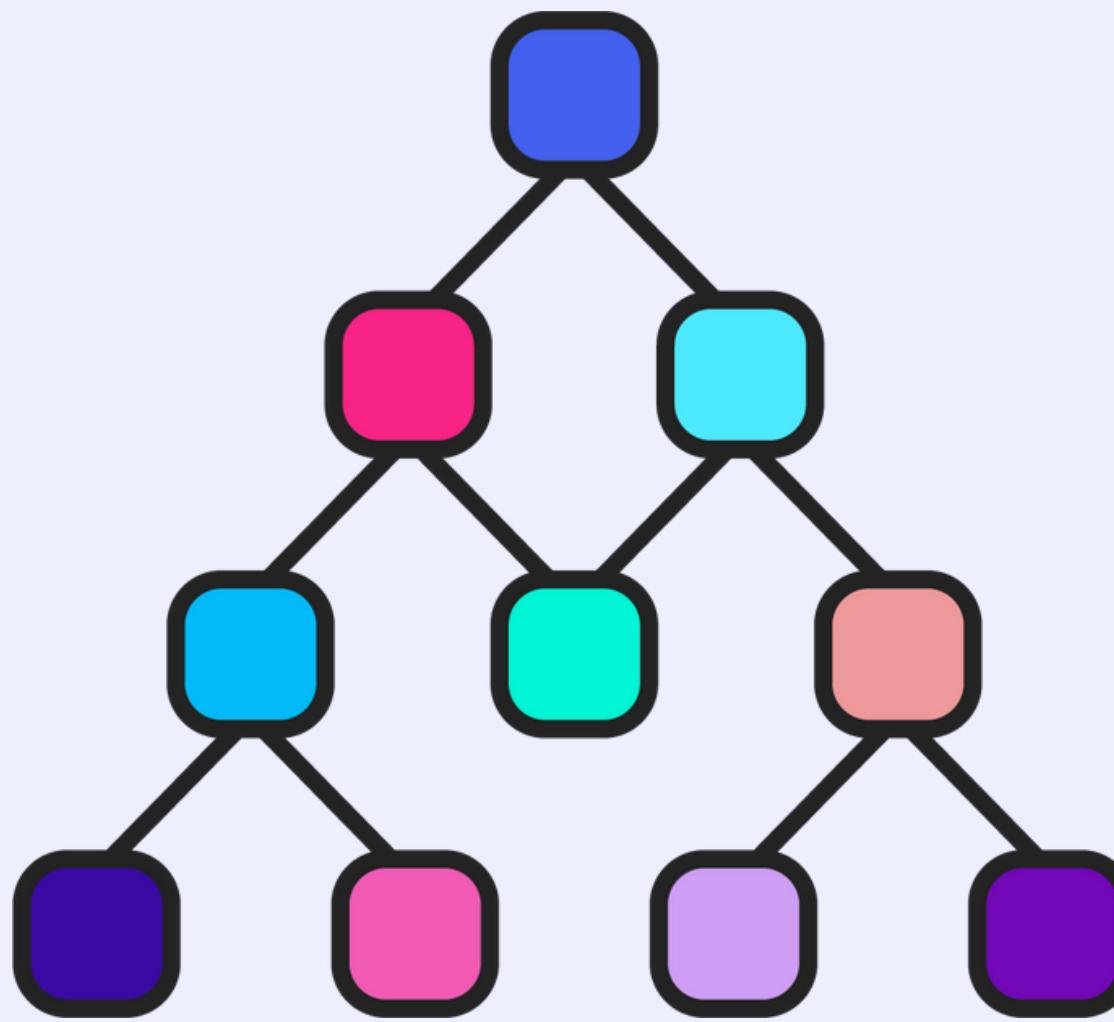
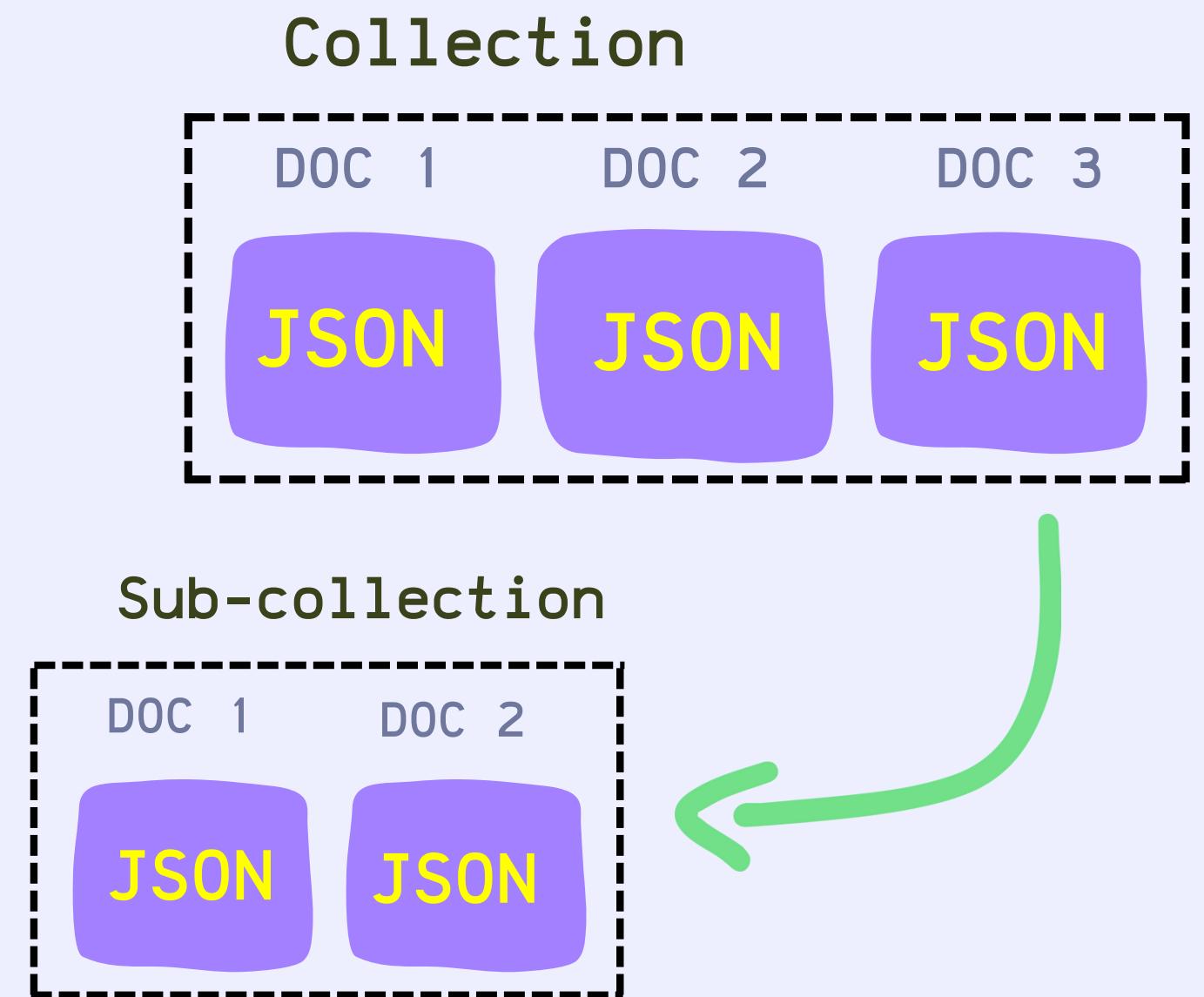
Cassandra



# DOCUMENT



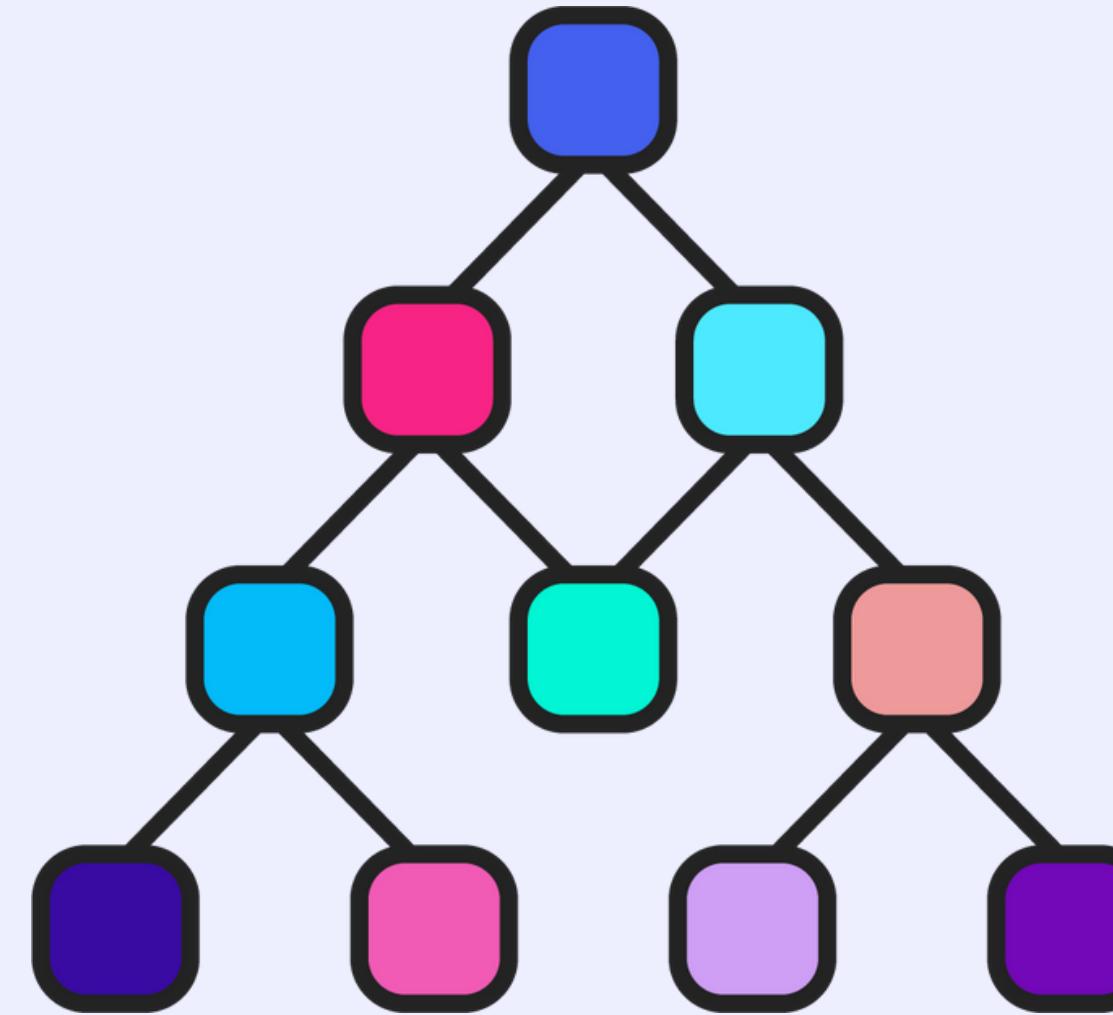
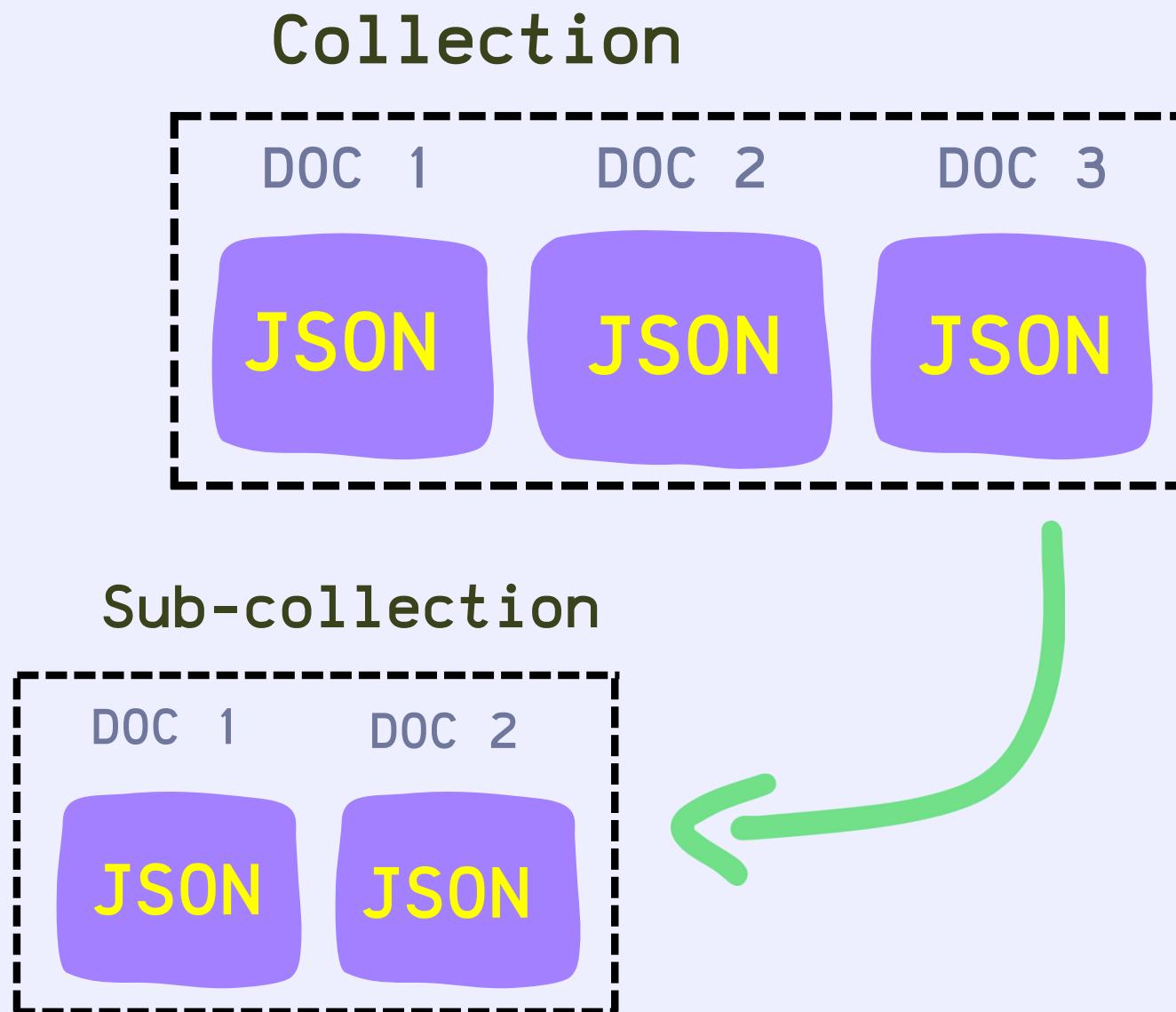
# DOCUMENT



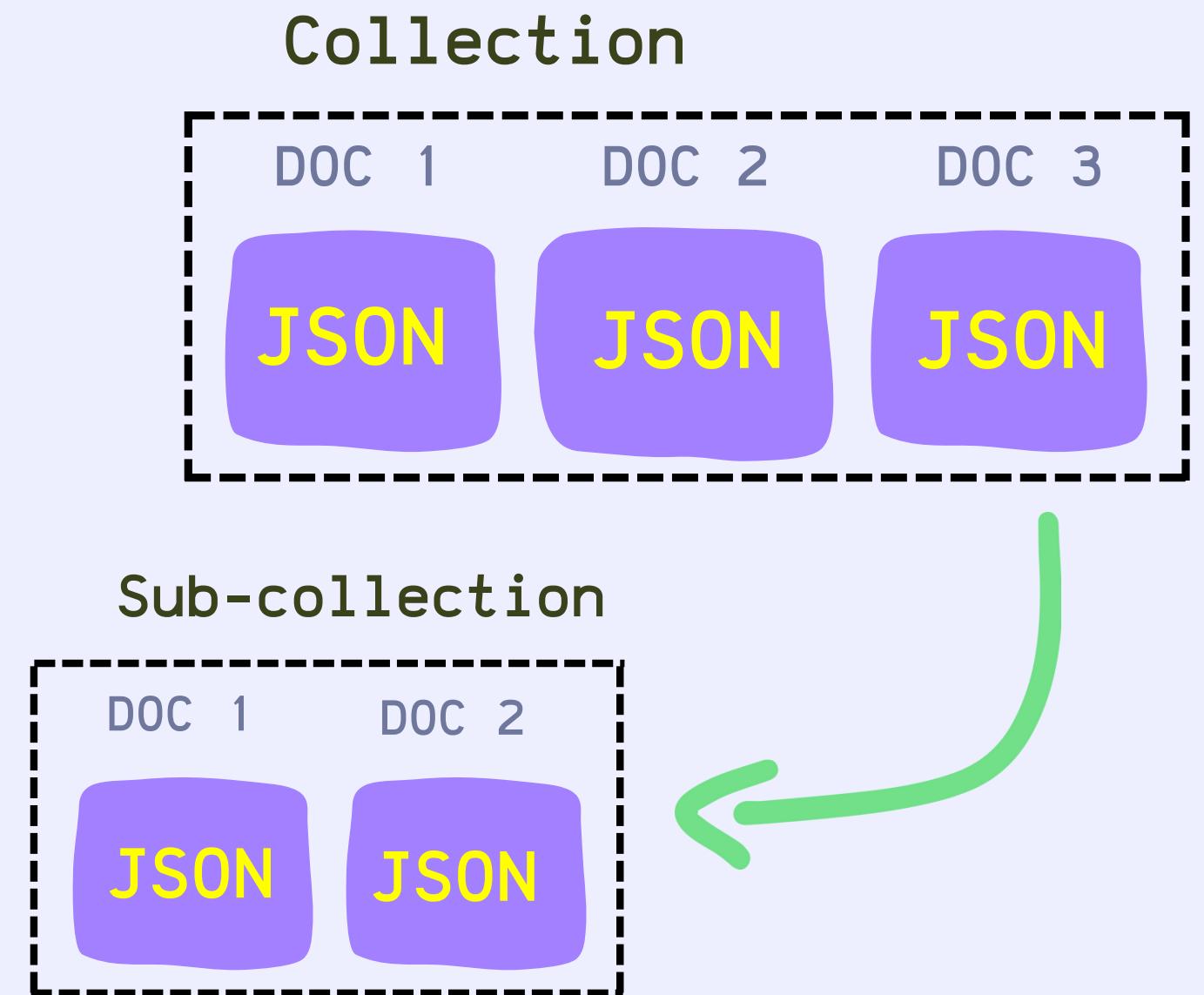
# DOCUMENT



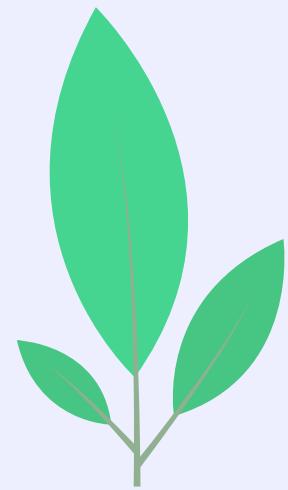
READS > WRITES



# DOCUMENT



DynamoDB



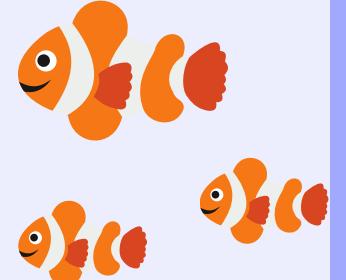
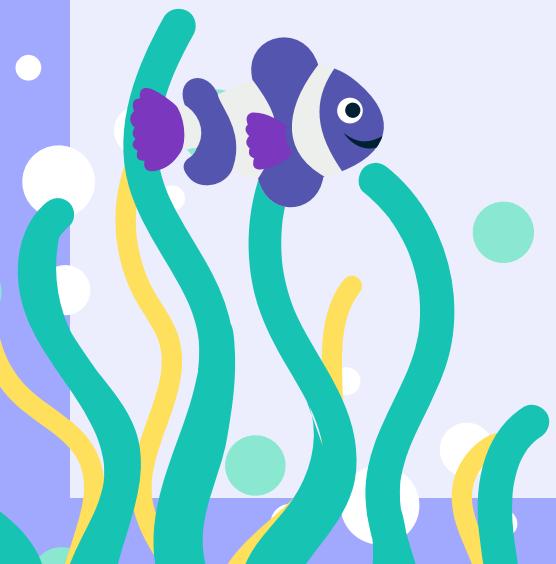
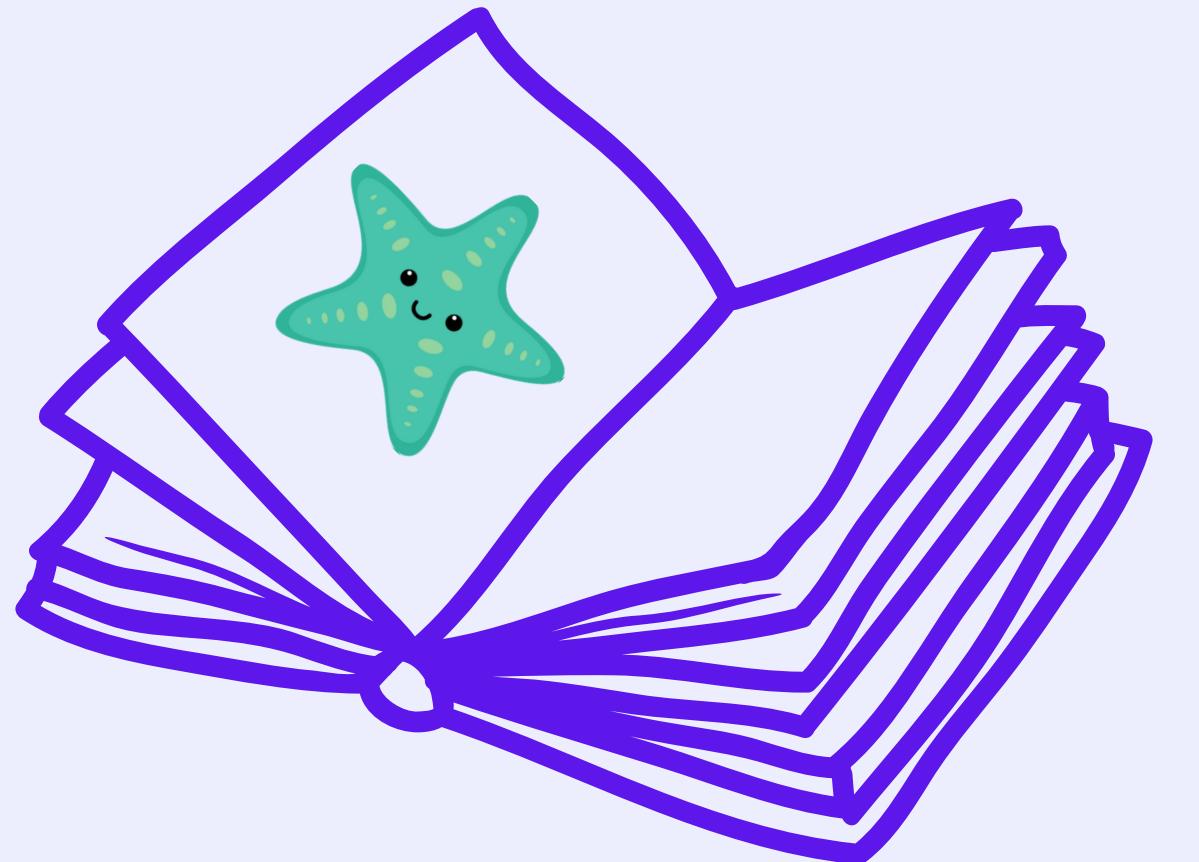
MongoDB



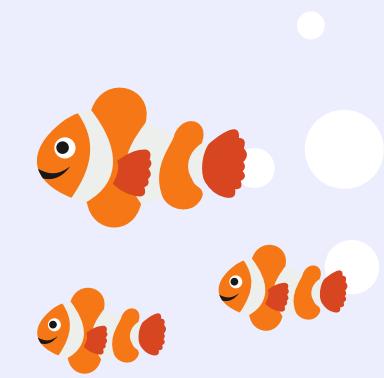
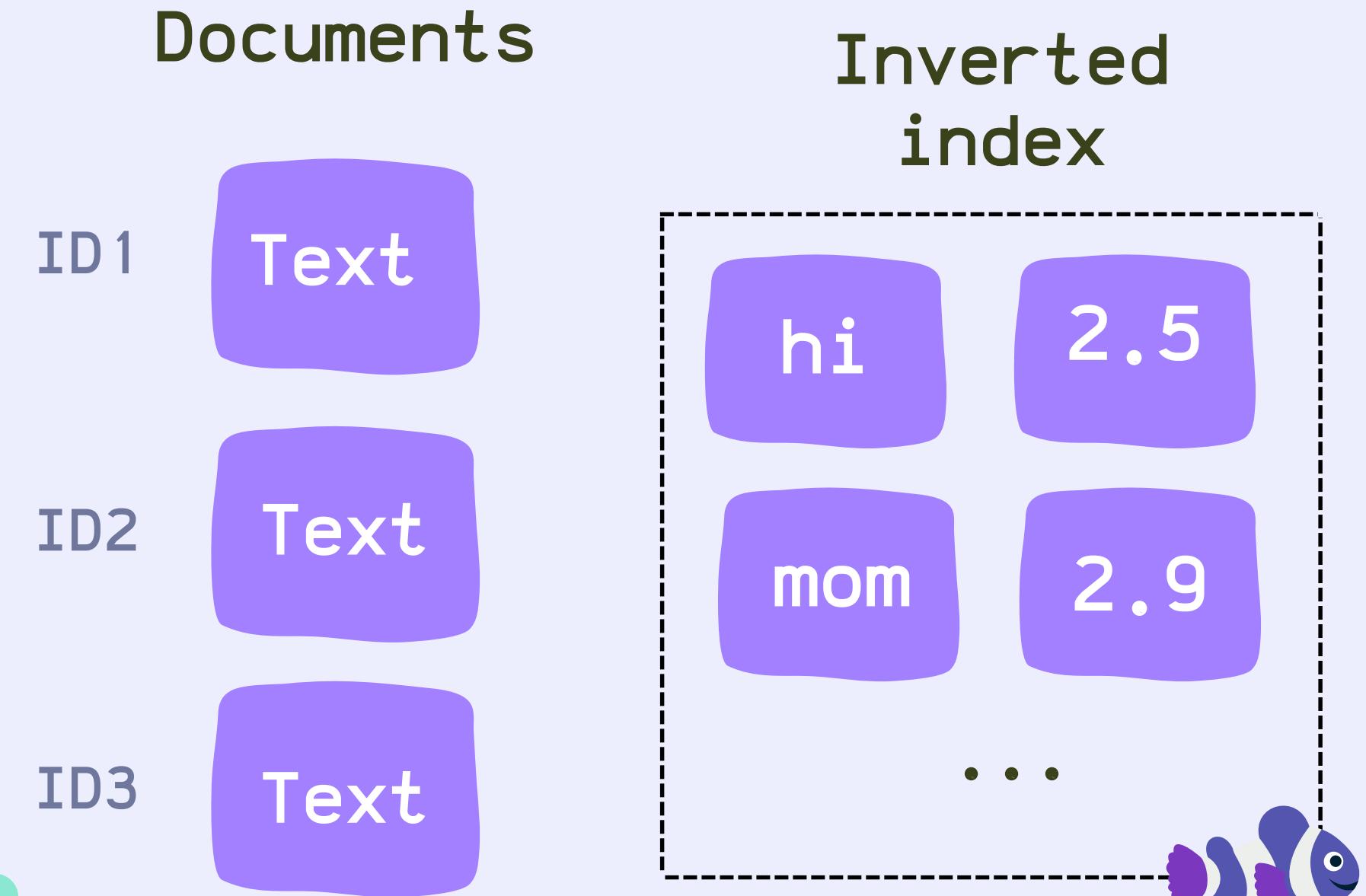
OpenSearch



# SEARCH

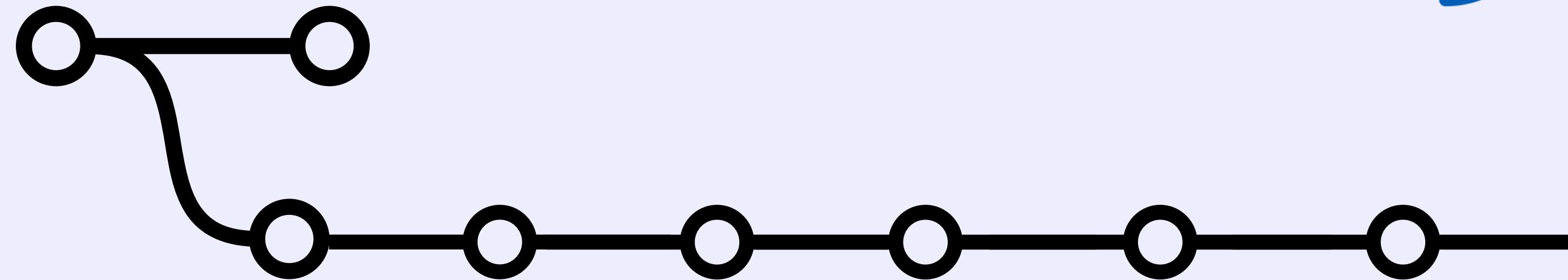


# SEARCH



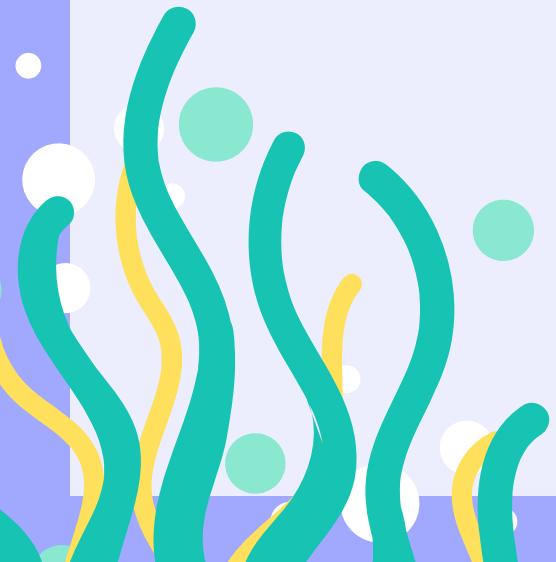
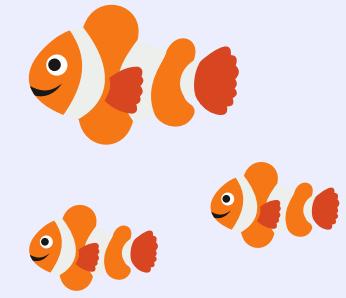
# SEARCH

Elasticsearch v7.10



OpenSearch v1.0.0

v2.2.1

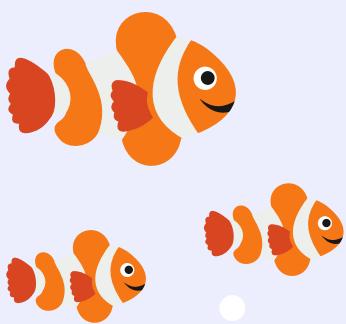
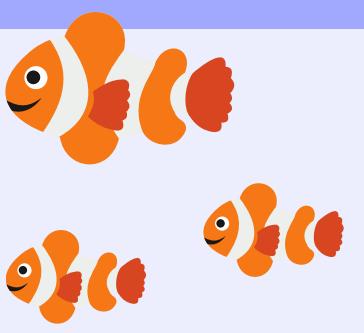


Multilingual

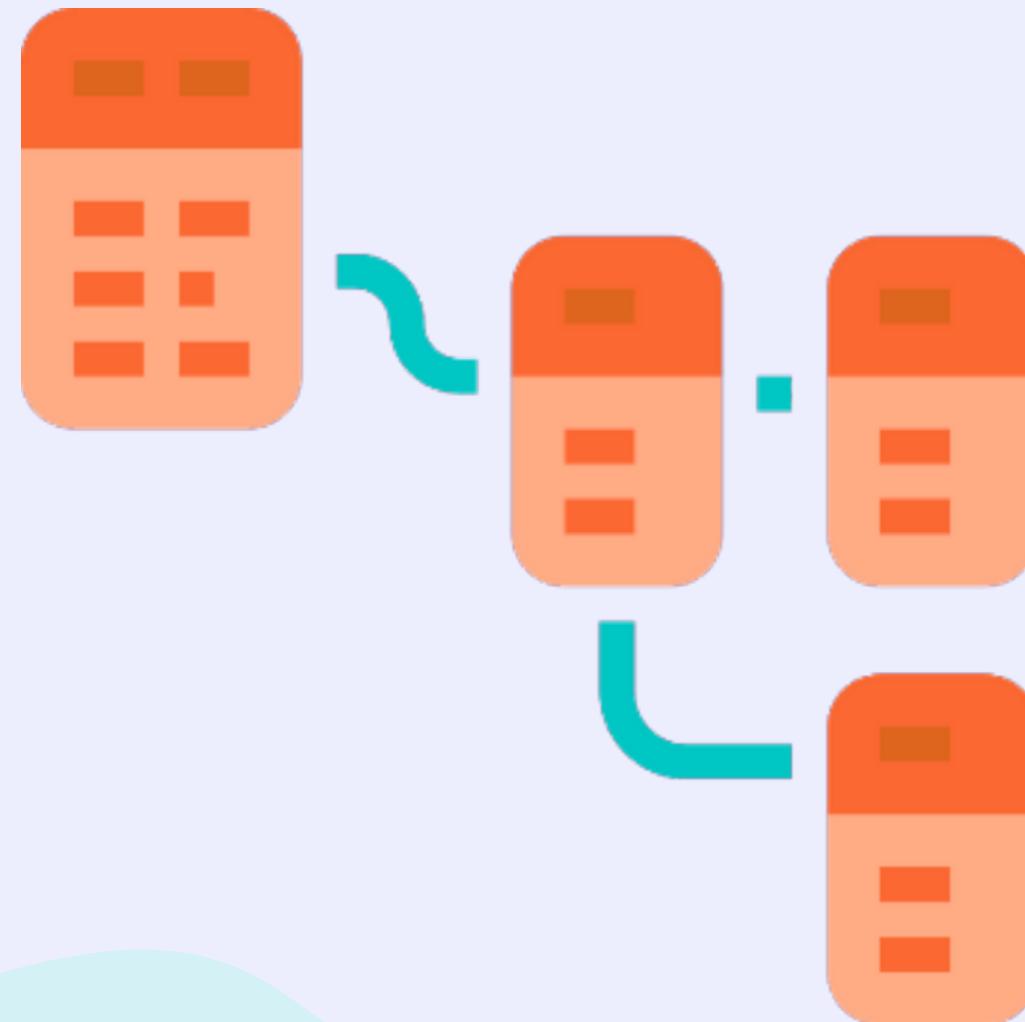
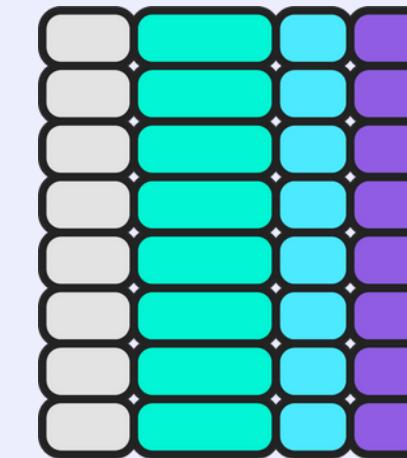
Autocompletion

Fuzzy searching

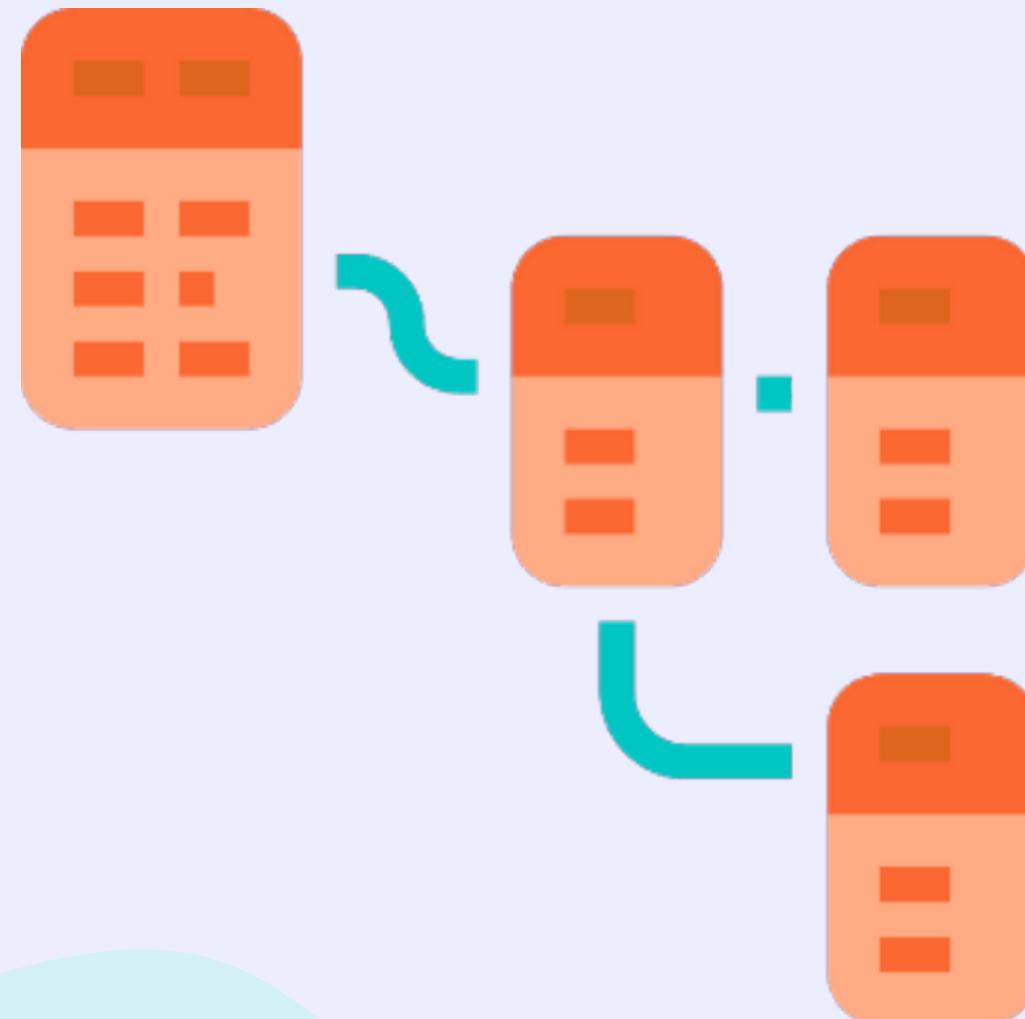
OSS



# RELATIONAL

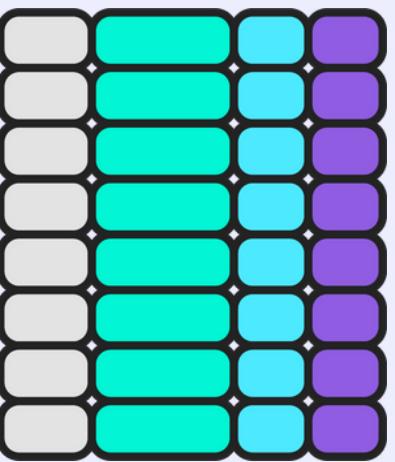
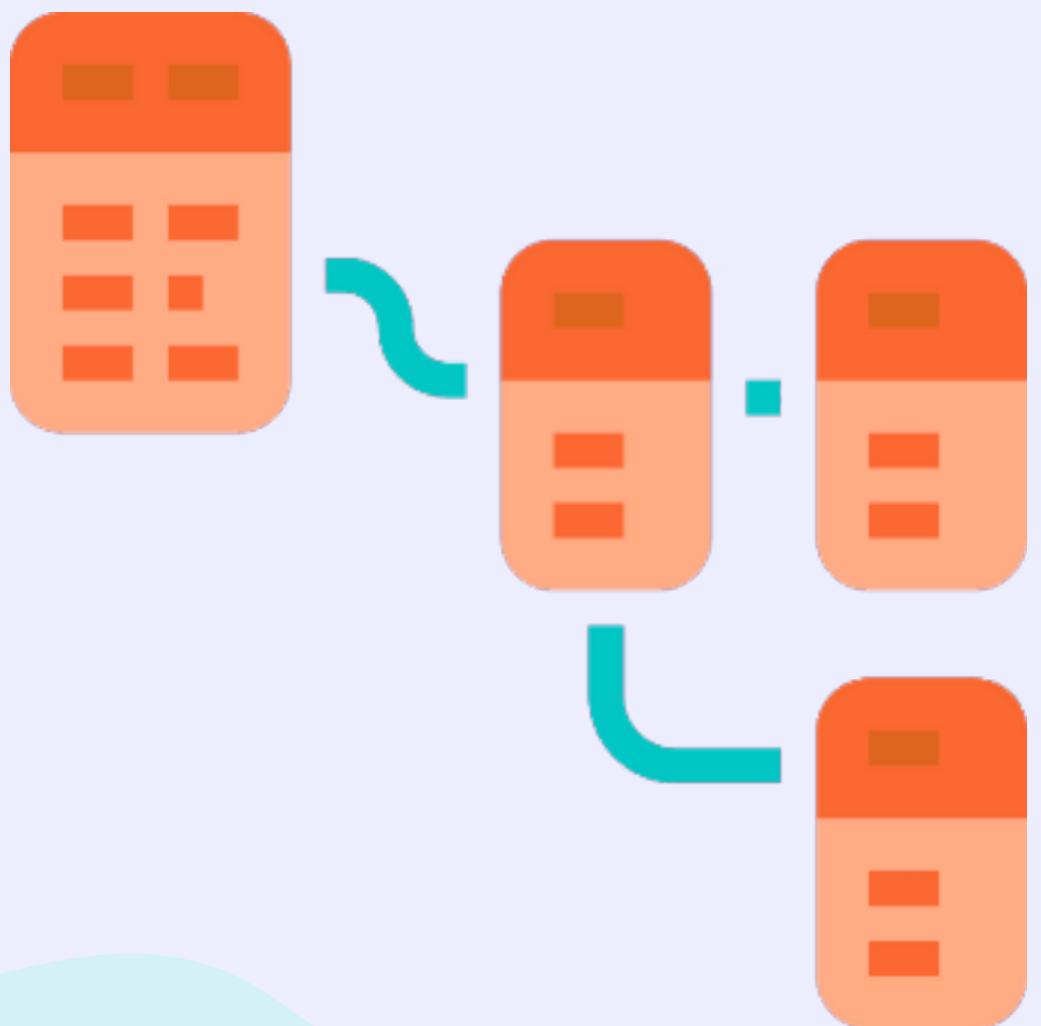


# RELATIONAL



PostgreSQL

# RELATIONAL



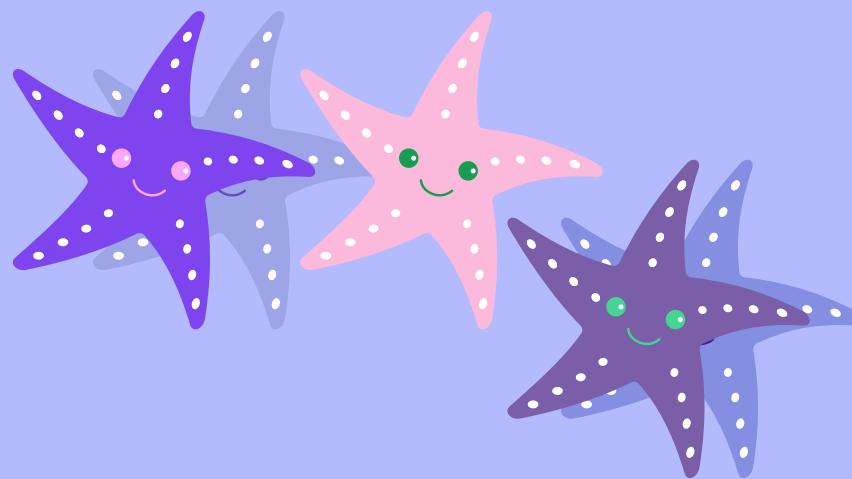
MySQL



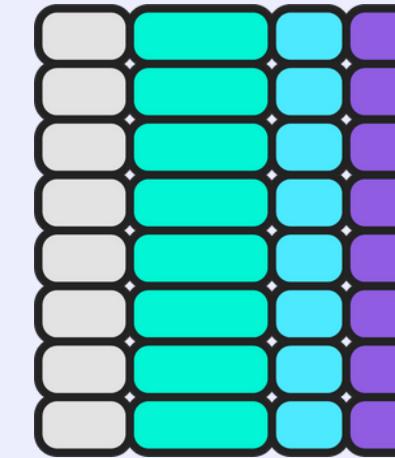
MariaDB

# ACID

**atomicity, consistency, isolation, durability**



# RELATIONAL

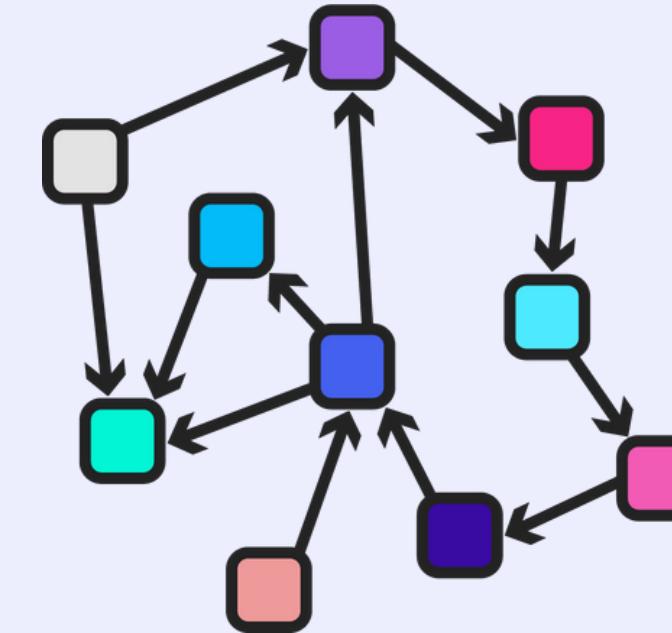
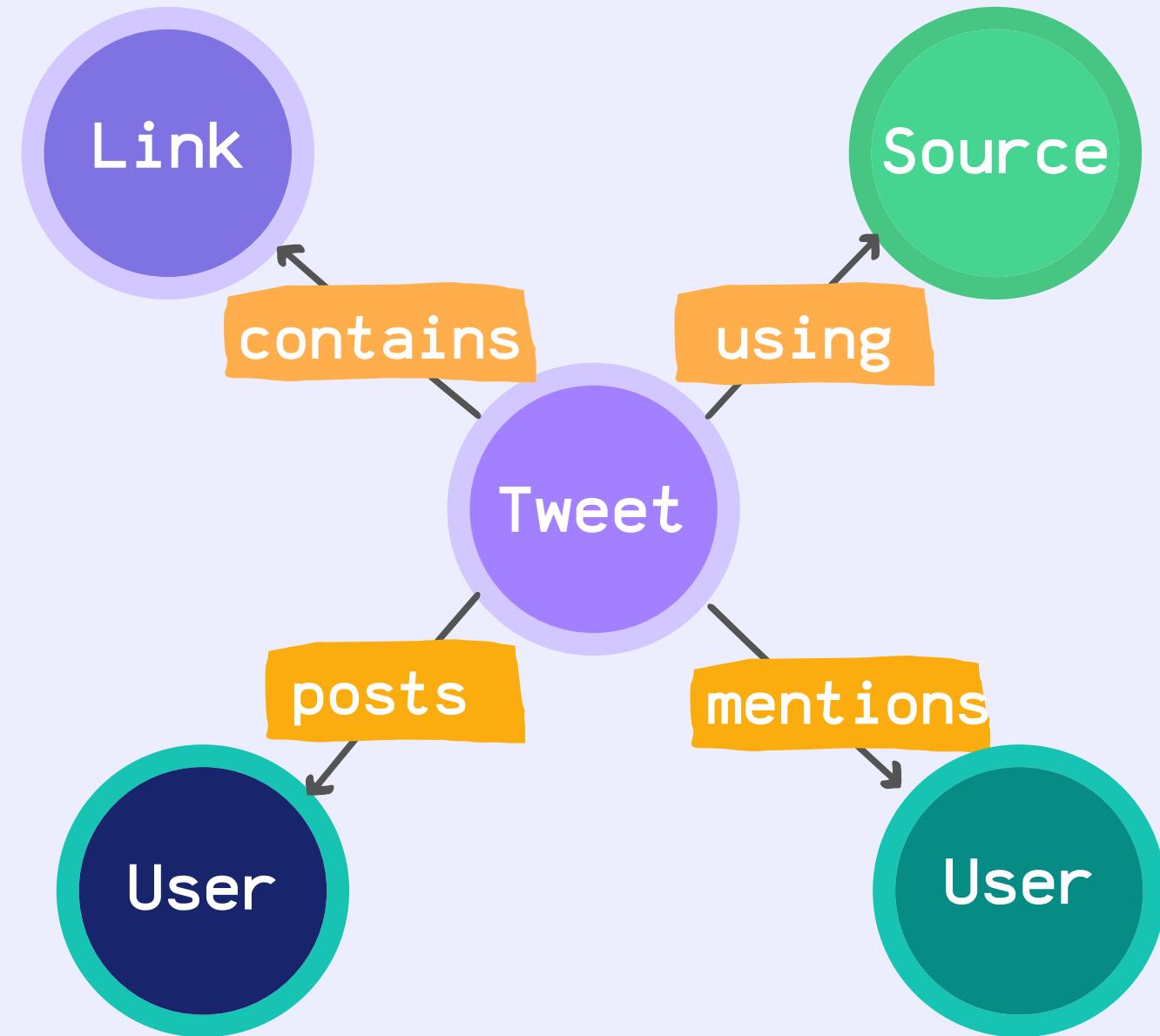


Schema

Lots of  
plugins

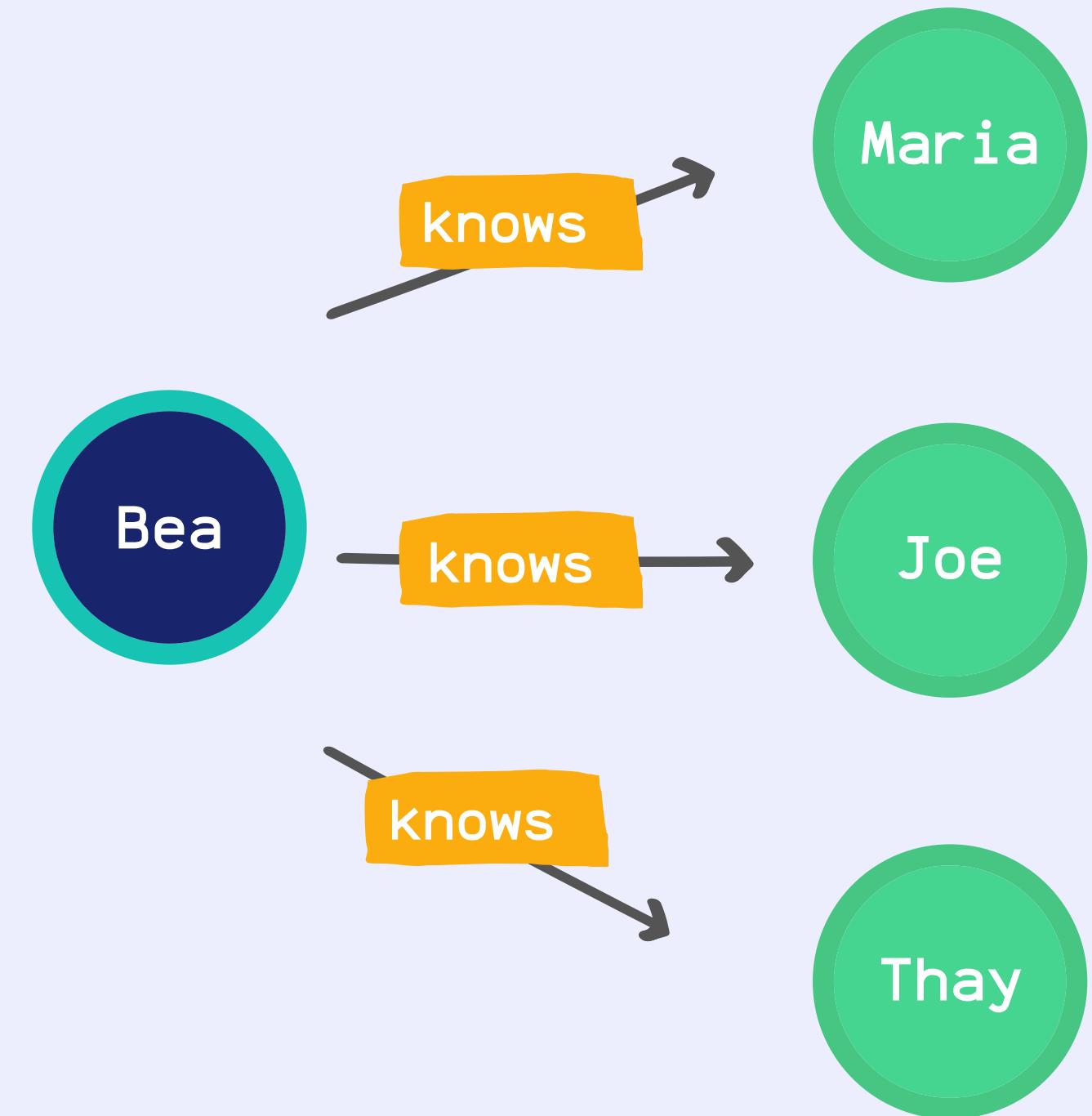
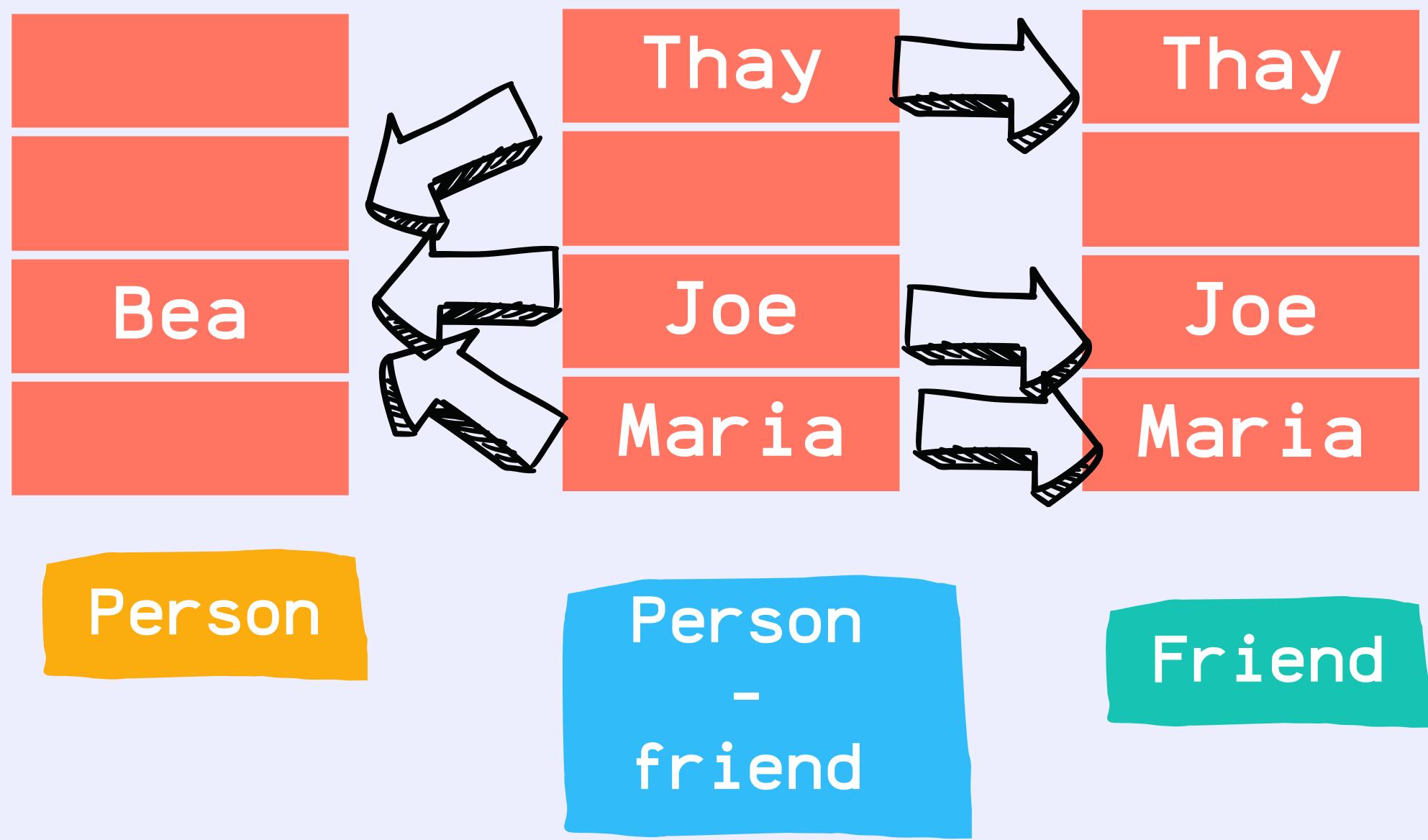
Documentation

# GRAPH



 neo4j

# GRAPH



# THOUGHTS



**WHAT YOU WANT !=  
WHAT YOU NEED!**



by @laysauchoa

# PLENTY MORE FISH IN THE SEA



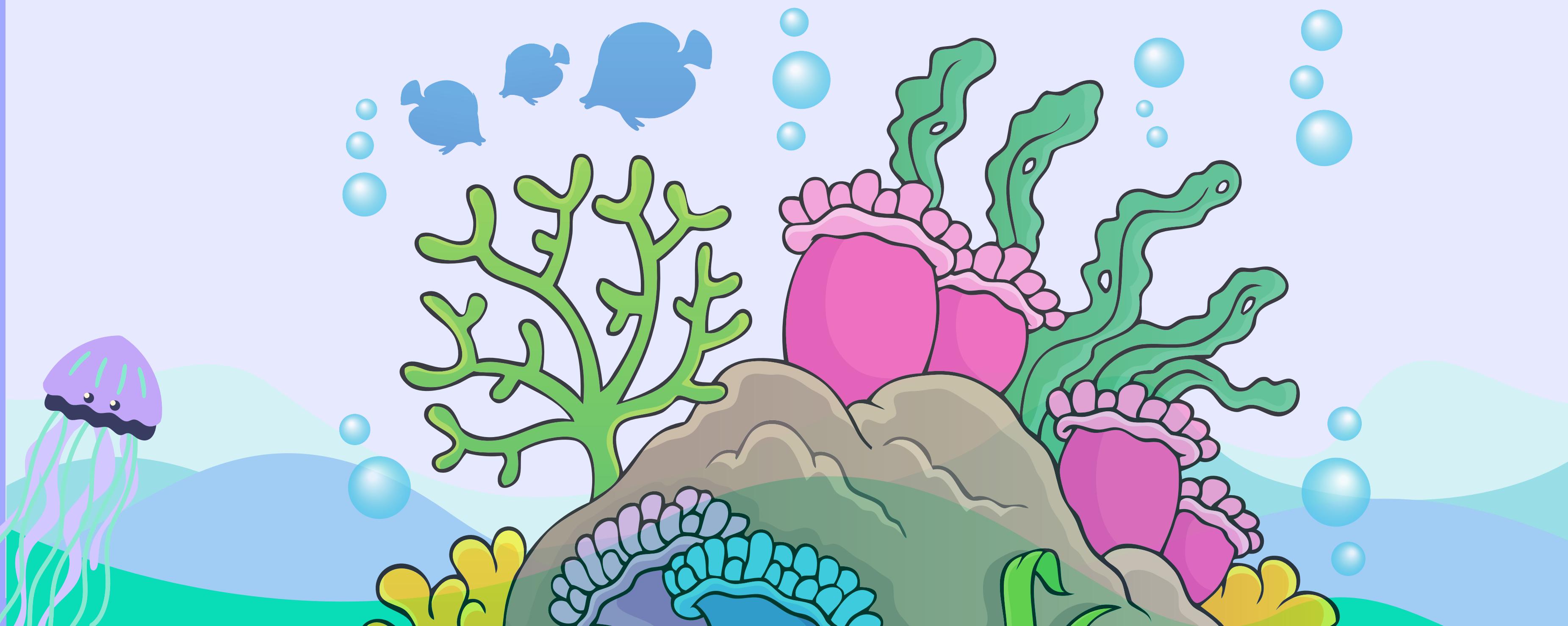
by @laysauchoa

# PLENTY MORE DB IN THE OPEN SEA

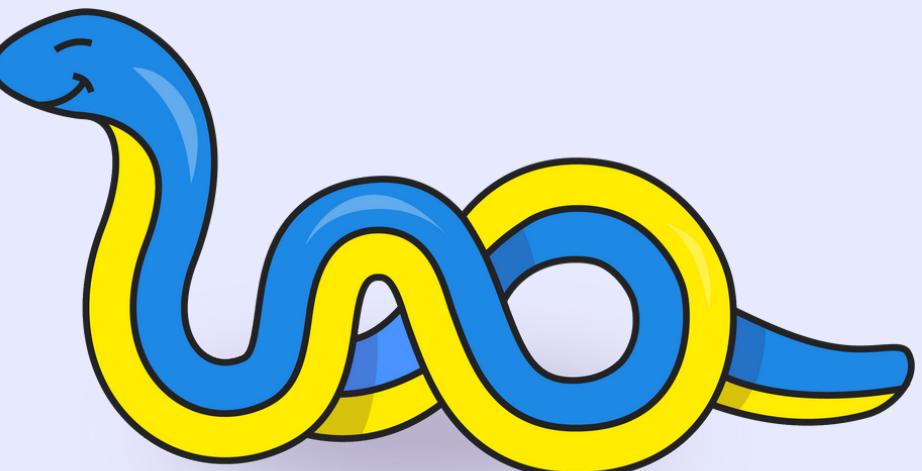


by @laysauchoa

# OPEN SOURCE



# THANK YOU!



@laysauchoa

@aiven\_io

