What‘s The Problem?

Up to this point, we typically stored our (simple) data in files

Create data

Replace entire content in file with new content

Read data

Always read entire file (= all the content)

Update data

Read entire content, find content to update, replace entire content

Delete data

Read entire content, find content to update, replace entire content

Pretty inefficient!

CRUD Operations

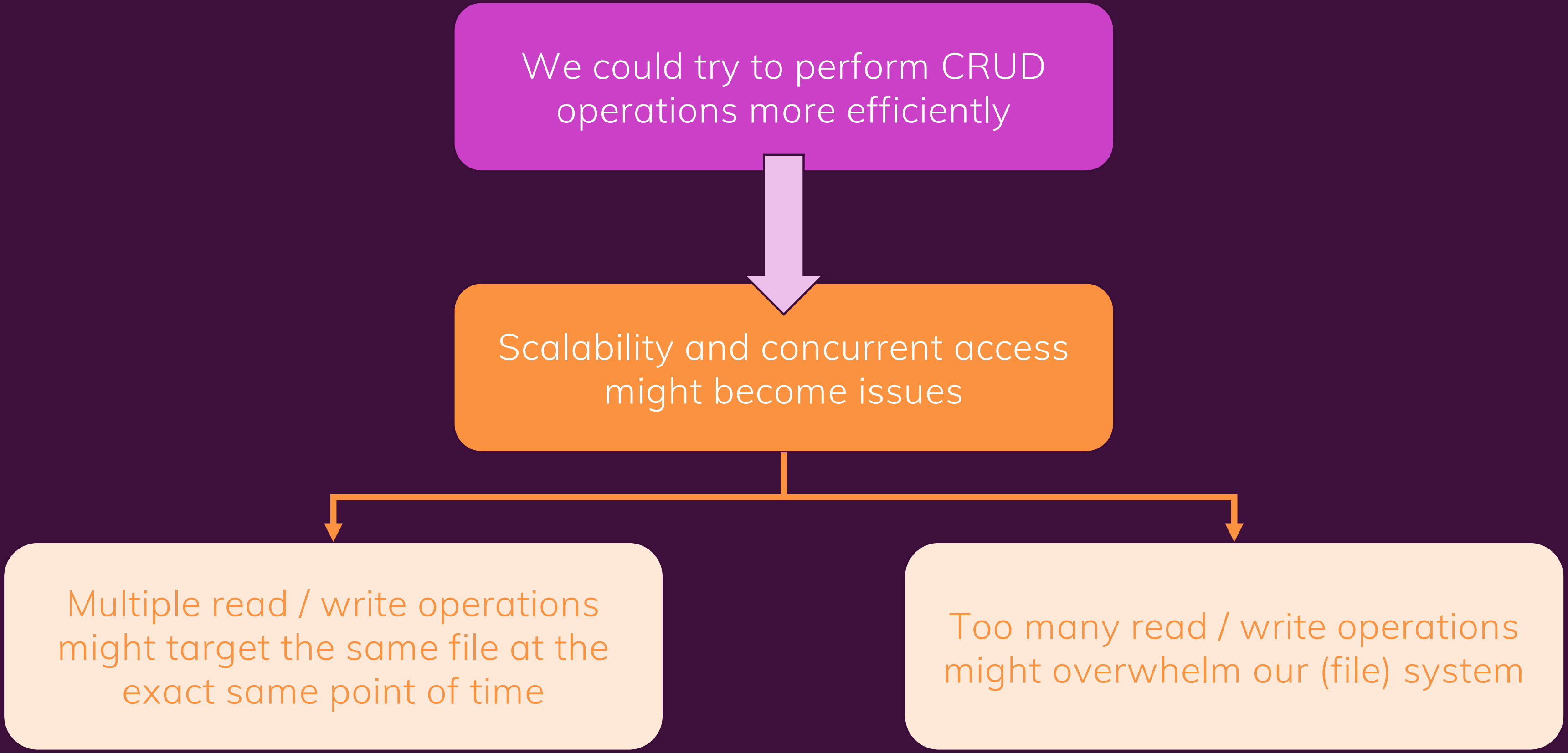
Create

Read

Update

Delete

That’s Not All!



We could try to perform CRUD operations more efficiently

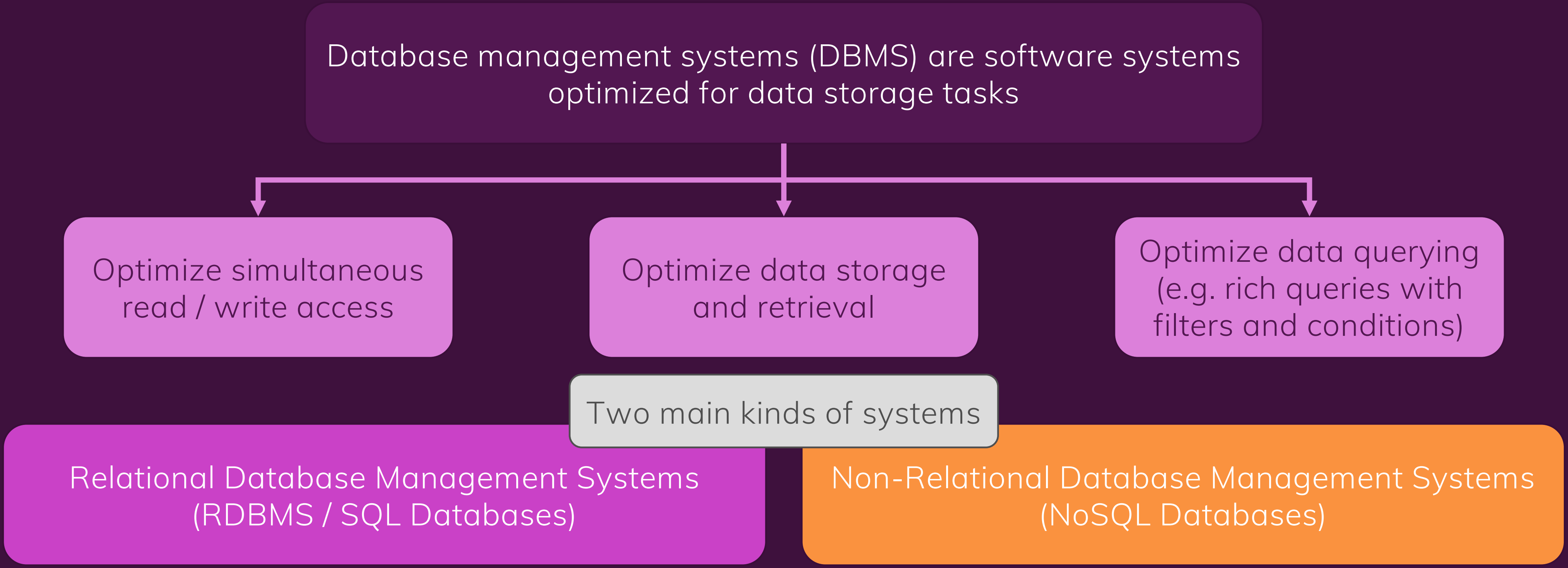
Scalability and concurrent access might become issues

Multiple read / write operations might target the same file at the exact same point of time

Too many read / write operations might overwhelm our (file) system

ACADE MIND

Database Systems To The Rescue!



Database management systems (DBMS) are software systems optimized for data storage tasks

Optimize simultaneous read / write access

Optimize data storage and retrieval

Optimize data querying (e.g. rich queries with filters and conditions)

Two main kinds of systems

Relational Database Management Systems (RDBMS / SQL Databases)

Non-Relational Database Management Systems (NoSQL Databases)

ACADEMIND

SQL

Structured Query Language

It actually was originally called “Sequel”

A Quick Introduction To SQL Databases

Store normalized data across multiple tables

Airports

ID	City	Country
MUC	Munich	Germany
JFK	New York	US
BCN	Barcelona	Spain

Flights

ID	Start	Dest
FL123	MUC	JFK
FL331	BCN	MUC
FL591	JFK	BCN

ACADE MIND

A Quick Introduction To SQL Databases

Airports

ID	City	Country
MUC	Munich	Germany
JFK	New York	US
BCN	Barcelona	Spain
Unique string	String	String

Flights

ID	Start	Dest
123	MUC	JFK
331	BCN	MUC
591	JFK	BCN
Unique integer	String	String

Tables have clearly defined schemas and data types

A Quick Introduction To SQL Databases

Airports

ID	City	Country
MUC	Munich	Germany
JFK	New York	US
BCN	Barcelona	Spain

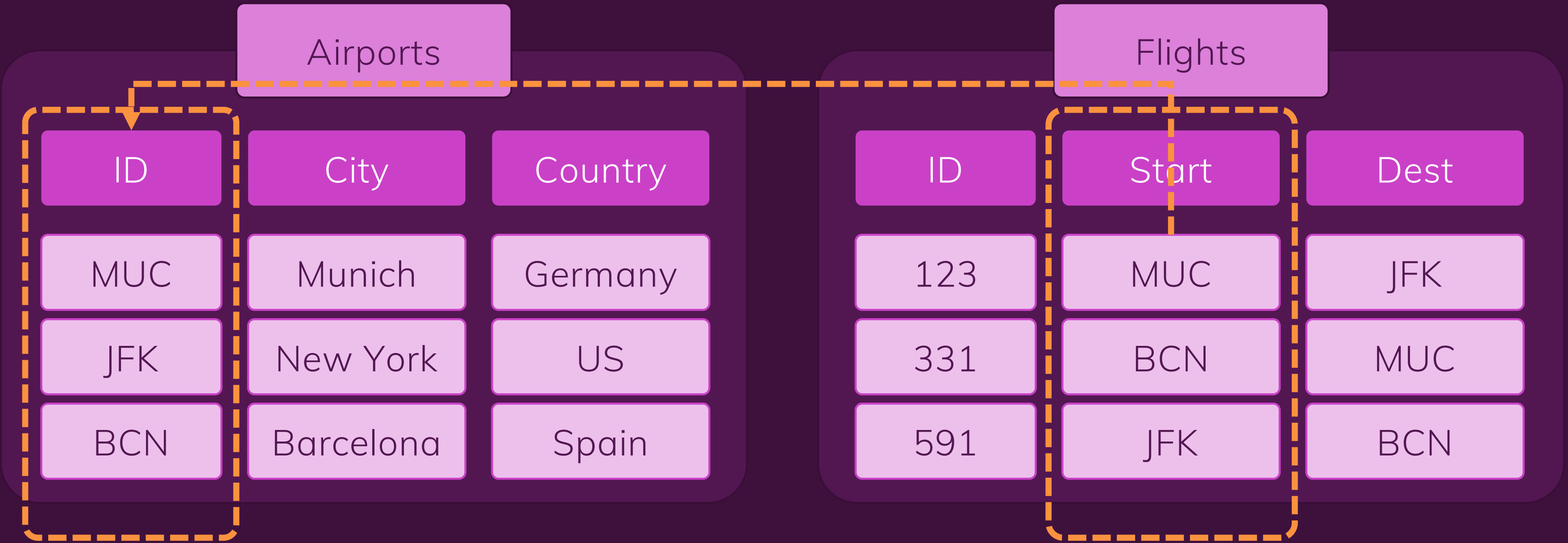
Flights

ID	Start	Dest
123	MUC	JFK
331	BCN	MUC
591	JFK	BCN

"Get all flights that start in MUC"

Data and relations can be queried

A Quick Introduction To SQL Databases



Airports

ID	City	Country
MUC	Munich	Germany
JFK	New York	US
BCN	Barcelona	Spain

Flights

ID	Start	Dest
123	MUC	JFK
331	BCN	MUC
591	JFK	BCN

"Get all flights that start in MUC and also get all the related airport data"

Data and relations can be queried

A Quick Introduction To NoSQL Databases

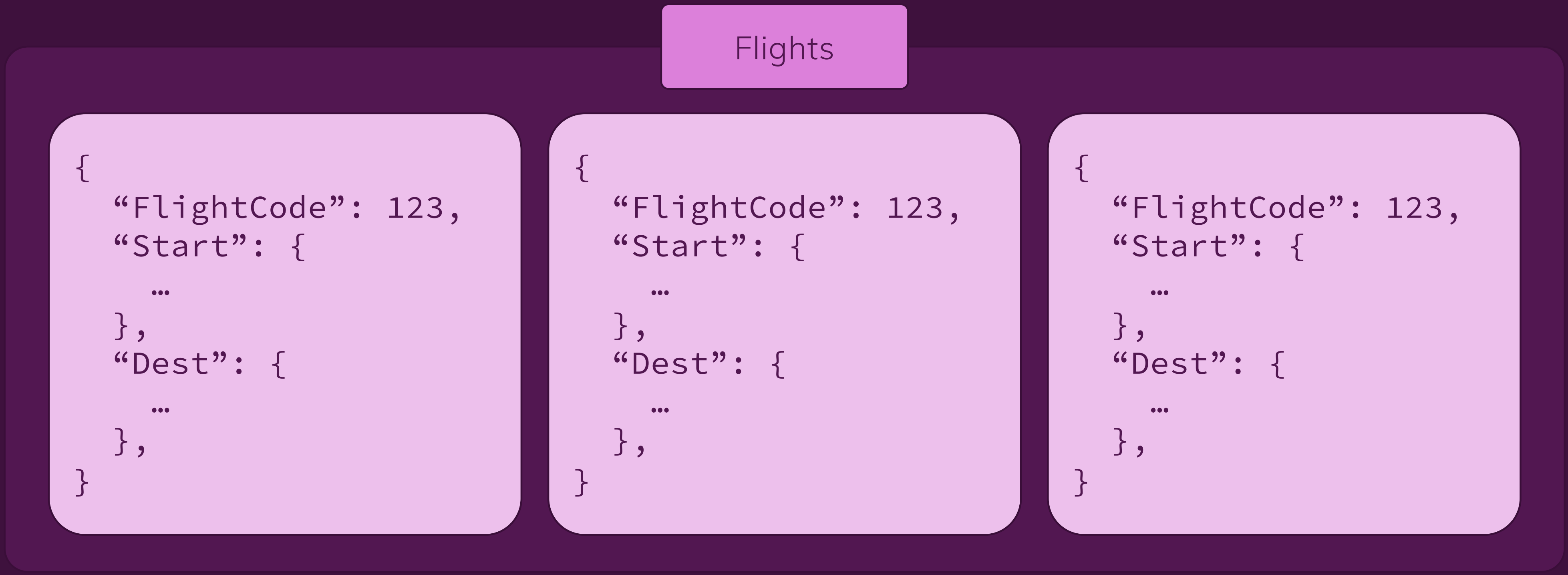
Flights

```
{
  “FlightCode”: 123,
  “Start”: {
    “APCode”: ”MUC”,
    “APCity”: “Munich”,
    “APCountry”: “Germany”
  },
  “Dest”: {
    “APCode”: ”JFK”,
    “APCity”: “New York”,
    “APCountry”: “US”
  },
}
```

```
{
  “FlightCode”: 331,
  “Start”: {
    “APCode”: ”BCN”,
    “APCity”: “Barcelona”,
    “APCountry”: “Spain”
  },
  “Dest”: {
    “APCode”: ”MUC”,
    “APCity”: “Munich”,
    “APCountry”: “Germany”
  },
}
```

```
{
  “FlightCode”: 591,
  “Start”: {
    “APCode”: ”JFK”,
    “APCity”: “New York”,
    “APCountry”: “US”
  },
  “Dest”: {
    “APCode”: ”BCN”,
    “APCity”: “Barcelona”,
    “APCountry”: “Spain”
  },
}
```

A Quick Introduction To NoSQL Databases



```
{
  “FlightCode”: 123,
  “Start”: {
    …
  },
  “Dest”: {
    …
  },
}
```

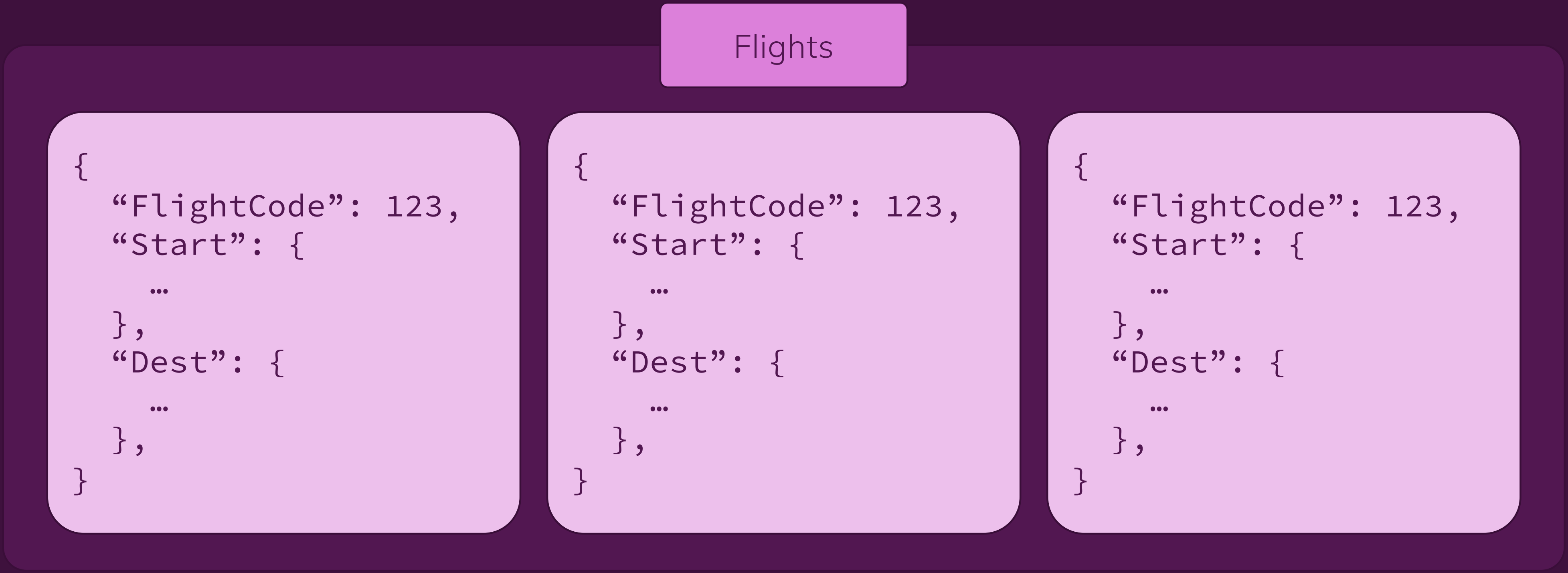
```
{
  “FlightCode”: 123,
  “Start”: {
    …
  },
  “Dest”: {
    …
  },
}
```

```
{
  “FlightCode”: 123,
  “Start”: {
    …
  },
  “Dest”: {
    …
  },
}
```

Data is stored in only a few tables which each contain more information

ACADE MIND

A Quick Introduction To NoSQL Databases



```
{
  "FlightCode": 123,
  "Start": {
    …
  },
  "Dest": {
    …
  },
}
```

```
{
  "FlightCode": 123,
  "Start": {
    …
  },
  "Dest": {
    …
  },
}
```

```
{
  "FlightCode": 123,
  "Start": {
    …
  },
  "Dest": {
    …
  },
}
```

More data can be fetched with fewer queries

ACADE
MIND

SQL vs NoSQL – Which One Should You Choose?

There is no clear winner

↓

Either system can be used and work for a given use-caes

↓

You should think about the queries you'll be running

SQL databases provide more structure and rules

NoSQL databases can be more flexible and reduce amount of require queries

Scalability can become an issue with SQL databases