Types of Databases

Databases, in general, are categorized into `Relational Databases` and `Non-Relational Databases`. Only Relational Databases utilize SQL, while Non-Relational databases utilize a variety of methods for communications.

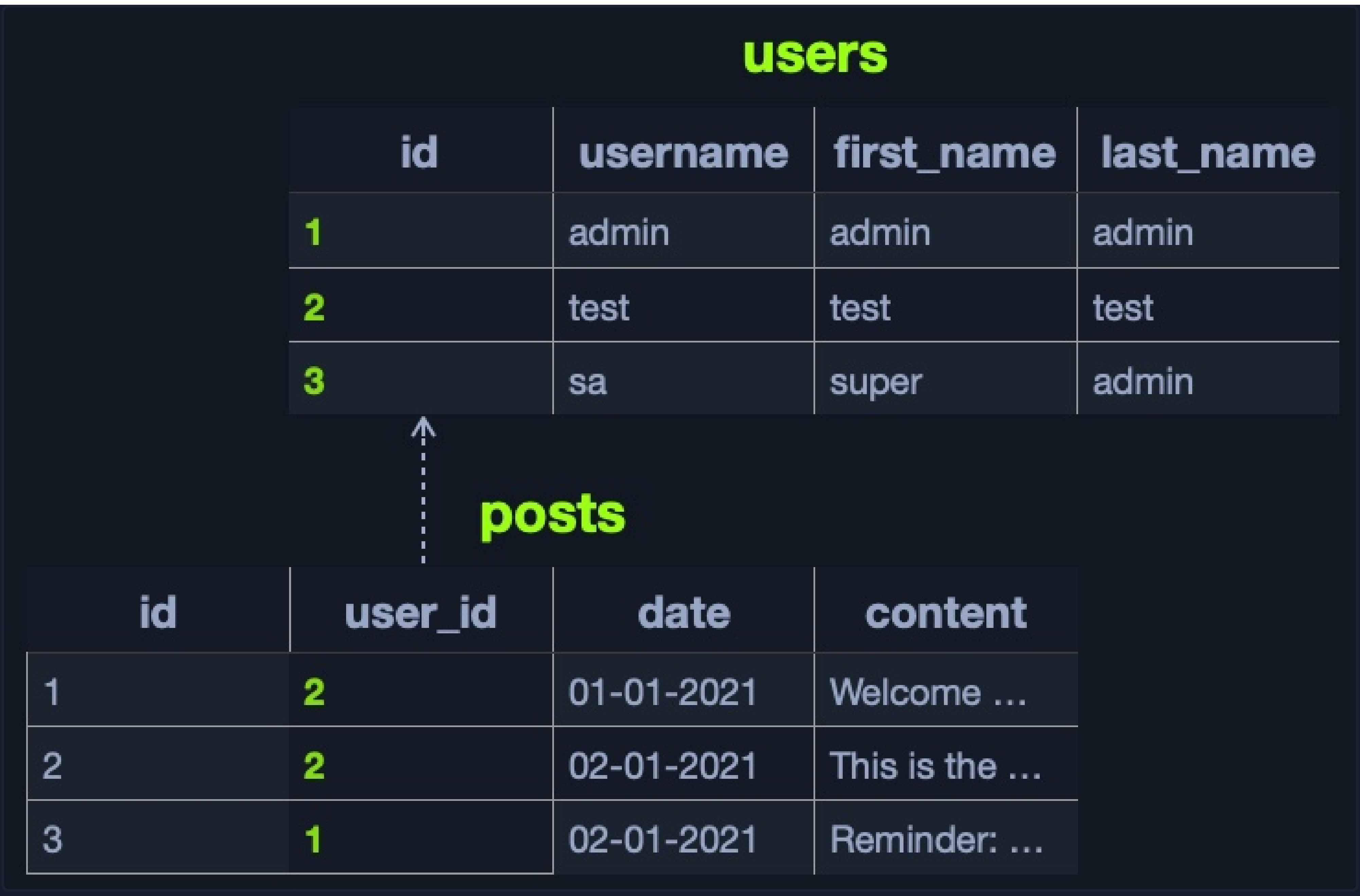
Relational Databases

A relational database is the most common type of database. It uses a schema, a template, to dictate the data structure stored in the database. For example, we can imagine a company that sells products to its customers having some form of stored knowledge about where those products go, to whom, and in what quantity. However, this is often done in the back-end and without obvious informing in the front-end. Different types of relational databases can be used for each approach. For example, the first table can store and display basic customer information, the second the number of products sold and their cost, and the third table to enumerate who bought those products and with what payment data.

Tables in a relational database are associated with keys that provide a quick database summary or access to the specific row or column when specific data needs to be reviewed. These tables, also called entities, are all related to each other. For example, the customer information table can provide each customer with a specific ID that can indicate everything we need to know about that customer, such as an address, name, and contact information. Also, the product description table can assign a specific ID to each product. The table that stores all orders would only need to record these IDs and their quantity. Any change in these tables will affect all of them but predictably and systematically.

However, when processing an integrated database, a concept is required to link one table to another using its key, called a `relational database management system` (`RDBMS`). Many companies that initially use different concepts are switching to the RDBMS concept because this concept is easy to learn, use and understand. Initially, this concept was used only by large companies. However, many types of databases now implement the RDBMS concept, such as Microsoft Access, MySQL, SQL Server, Oracle, PostgreSQL, and many others.

For example, we can have a `users` table in a relational database containing columns like `id`, `username`, `first_name`, `last_name`, and others. The `id` can be used as the table key. Another table, `posts`, may contain posts made by all users, with columns like `id`, `user_id`, `date`, `content`, and so on.



users

id	username	first_name	last_name
1	admin	admin	admin
2	test	test	test
3	sa	super	admin

posts

id	user_id	date	content
1	2	01-01-2021	Welcome ...
2	2	02-01-2021	This is the ...
3	1	02-01-2021	Reminder: ...

We can link the `id` from the `users` table to the `user_id` in the `posts` table to retrieve the user details for each post without storing all user details with each post. A table can have more than one key, as another column can be used as a key to link with another table. So, for example, the `id` column can be used as a key to link the `posts` table to another table containing comments, each of which belongs to a particular post, and so on.

The relationship between tables within a database is called a Schema.

This way, by using relational databases, it becomes rapid and easy to retrieve all data about a particular element from all databases. So, for example, we can retrieve all details linked to a specific user from all tables with a single query. This makes relational databases very fast and reliable for big datasets with clear structure and design and efficient data management. The most common example of relational databases is `MySQL`, which we will be covering in this module.

Non-relational Databases

A non-relational database (also called a `NoSQL` database) does not use tables, rows, and columns or prime keys, relationships, or schemas. Instead, a NoSQL database stores data using various storage models, depending on the type of data stored. Due to the lack of a defined structure for the database, NoSQL databases are very scalable and flexible. Therefore, when dealing with datasets that are not very well defined and structured, a NoSQL database would be the best choice for storing such data. There are four common storage models for NoSQL databases:

- `Key-Value`
- `Document-Based`
- `Wide-Column`
- `Graph`

Each of the above models has a different way of storing data. For example, the `Key-Value` model usually stores data in JSON or XML, and have a key for each pair, and stores all of its data as its value:

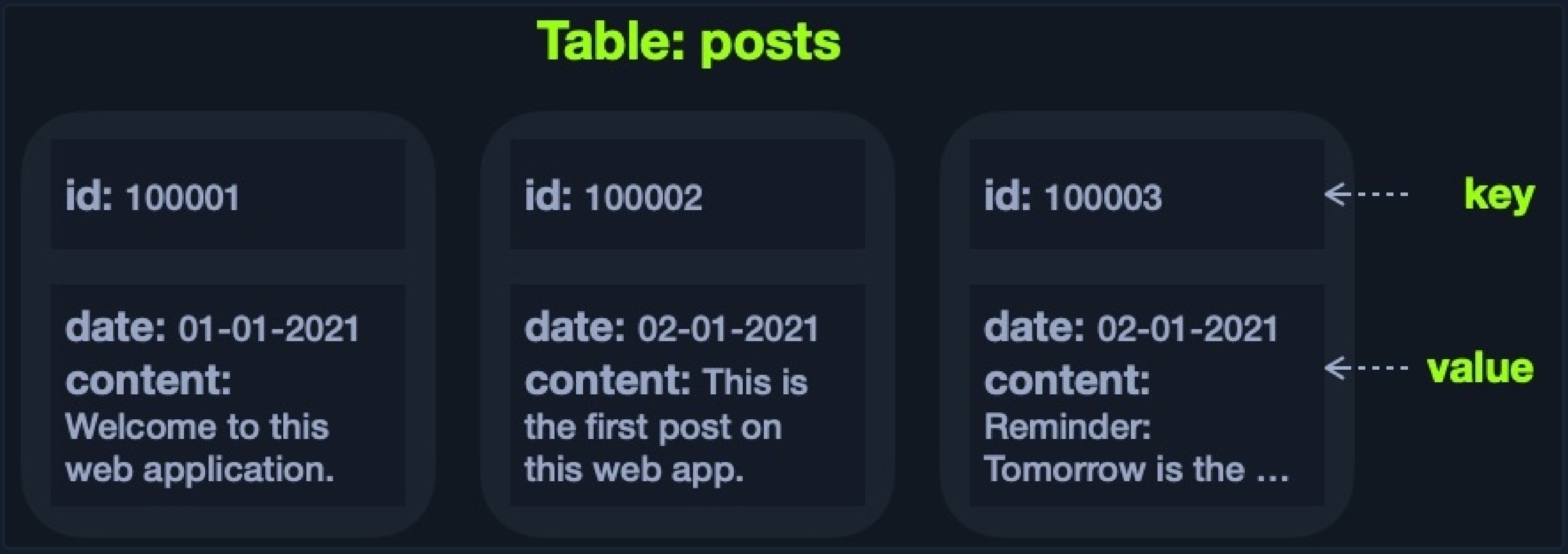


Table: posts
id: 100001
date: 01-01-2021 content: Welcome to this web application.
id: 100002
date: 02-01-2021 content: This is the first post on this web app.
id: 100003
date: 02-01-2021 content: Reminder: Tomorrow is the ...
key
value

The above example can be represented using JSON as:

Code: json

```json
{
  "100001": {
    "date": "01-01-2021",
    "content": "Welcome to this web application."
  },
  "100002": {
    "date": "02-01-2021",
    "content": "This is the first post on this web app."
  },
  "100003": {
    "date": "02-01-2021",
    "content": "Reminder: Tomorrow is the ..."
  }
}
```

It looks similar to a dictionary item in languages like `Python` or `PHP` (i.e. `{'key':'value'}`), where the `key` is usually a string, and the `value` can be a string, dictionary, or any class object.

The most common example of a NoSQL database is `MongoDB`.

Non-relational Databases have a different method for injection, known as NoSQL injections. SQL injections are completely different than NoSQL injections. NoSQL injections will be covered in a later module.

Previous Next Mark Complete & Next

Cheat Sheet

Table of Contents

My Workstation

O F F L I N E

Start Instance

1 / 1 spawns left