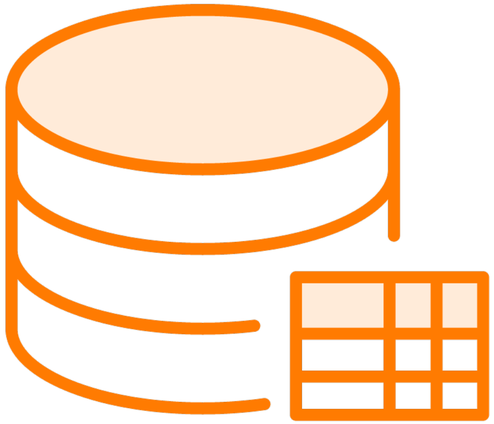
Comparing Databases

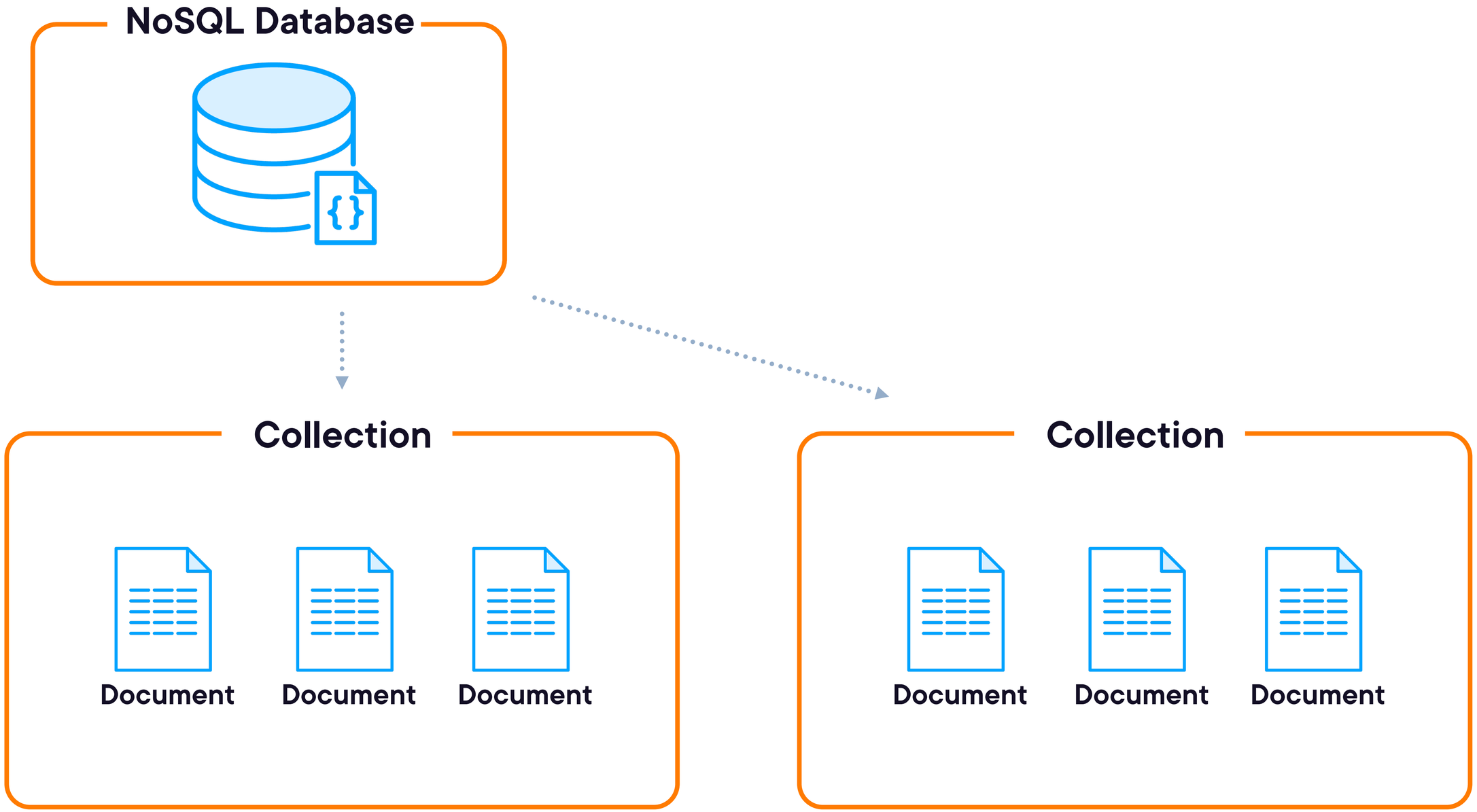
NoSQL/Non-Relational Database



SQL/Relational Database

Relational Database: Tables, Rows, Columns

profile_id	firstName	lastName	company
1	Jane	Smith	XYZ Inc
2	Michael	Johnson	123 Company
3	Sarah	Williams	ABC Corp


NoSQL Database
Collection
Document
Document
Document
Collection
Document
Document
Document

MongoDB Document, a JSON-like Structure

```
{
    "profile_id": 1,
    "first_name": “Jane",
    "last_name": “Smith",
    "company": “XYZ Inc”
},
{
    "profile_id": 2,
    "first_name": “Michael",
    "last_name": “Johnson",
    "company": “123 Company”
}
```

Nested Documents

```
{
  "post_id": 1,
  "profile_id": 2,
  "content": “My amazing first post!”,
  "tags": [“new-hire", "hobbies", “intro”]
}
```

Nested Documents

```
{
  "profile_id": 2,
  "first_name": "Jane",
  "last_name": "Smith",
  "company": "XYZ Inc",
  "posts": [
    {
      "post_id": 1,
      "content": “My amazing first post!”,
      "tags": [“new-hire", "hobbies", "intro"]
    }
  ]
}
```

SQL (Structured Query Language)

```sql
SELECT *
FROM profiles
WHERE name = 'My Mongo DB Document';
```

NoSQL = Not Only SQL

Querying MongoDB with JavaScript

app.js

```
const results = await collection.findOne({ name: “My MongoDB Document”});

console.log(results);
```

A Database Schema

A schema defines the structure and relationships in a database.

Defining a SQL Schema

```sql
CREATE TABLE profiles (
    profile_id INT PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    company VARCHAR(255)
);
```

MongoDB does not enforce a schema (but it's still probably a good idea).

Comparing Databases - Review

NoSQL/Non-relational	VS.	SQL/Relational
Unstructured, flexible data		Predefined, structured data
Document-based, nested data		Tables, rows, columns
Seamlessly use one language		Use SQL or ORM
Risk of inconsistency and difficult maintenance		Less flexibility

Let’s Go!