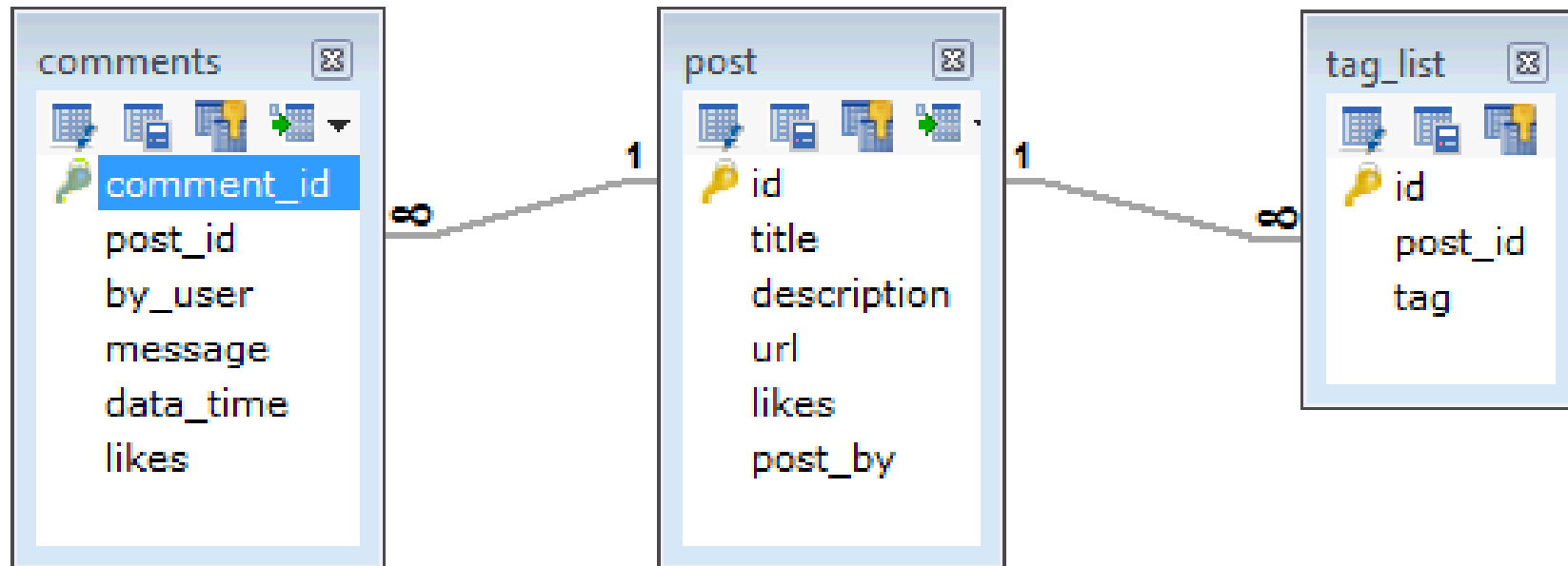


**Dr. Wittawin Susutti**

**wittawin.gas  
@mail.kmutt.ac.th**

# **CSS227 WEB PROGRAMMING LECTURE 09 – NOSQL**

# RDBMS schema



# What's NoSQL?

- A No SQL database provides a mechanism for storage and retrieval of data that employs **less constrained consistency models** than traditional relational database
- No SQL systems are also referred to as "Not-only-SQL".

# Characteristics of NoSQL

- NoSQL **avoids:**
  - Overhead of ACID transactions
  - Complexity of SQL query
  - Burden of up-front schema design
  - Transactions
- **Provides:**
  - Easy and frequent changes to DB
  - Fast development
  - Large data volumes
  - Schema less

# Schema-less datamodel

In relational Databases:

- You can't add a record which **does not fit the schema**
- You need to **add NULLs to unused items** in a row
- We should **consider the datatypes**.
  - i.e.. : you can't add a string to an integer field
- You **can't add multiple items in a field**

# Schema-less datamodel

In NoSQL Databases:

- There is **no schema** to consider
- There is **no unused cell**
- There is **no datatype** (implicit)
- We gather all items in a document

# What are the Types of NoSQL Databases?

- **Document databases**

- store data in documents similar to JSON (JavaScript Object Notation) objects
- Each document contains pairs of fields and values.

- **Key-value databases**

- a simpler type of database where each item contains keys and values.
- A value can typically only be retrieved by referencing its value

- **Wide-column stores**

- store data in tables, rows, and dynamic columns.
- each row is not required to have the same columns.

- **Graph databases**

- store data in nodes and edges.
- Nodes typically store information about people, places, and things while edges store information about the relationships between the nodes.

# NoSQL Databases

- **Pros**

- Can store arbitrary objects
- Easy to set up and use (little management)
- Easy to change
- Scales nicely

- **Cons**

- No transactional guarantees
- Data consistency is up to the user
- Complex queries need to be coded explicitly



# SQL vs NoSQL

Parameter	SQL	NOSQL
Definition	SQL databases are primarily called RDBMS or <b>Relational Databases</b>	NoSQL databases are primarily called as <b>Non-relational or distributed database</b>
Design for	Traditional RDBMS uses SQL syntax and queries to analyze and get the data for further insights.	These databases were developed in response to the demands presented for the development of the modern application.
Query Language	Structured query language ( <b>SQL</b> )	<b>No declarative query language</b>
Type	SQL databases are <b>table-based databases</b>	NoSQL databases can be <b>document based</b> , key-value pairs, graph databases
Schema	SQL databases have a <b>predefined schema</b>	NoSQL databases use <b>dynamic schema</b> for unstructured data.
Examples	Oracle, Postgres, and MS-SQL.	MongoDB, Redis, Neo4j, Cassandra, Hbase.
Best suited for	An ideal choice for the <b>complex query intensive environment</b> .	It is <b>not good fit complex queries</b> .
Best Used for	RDBMS database is the right option for <b>solving ACID problems</b> .	NoSQL is a best used for solving <b>data availability problems</b>
Importance	It should be used when <b>data validity</b> is super important	Use when it's more important to have <b>fast data than correct data</b>
ACID vs. BASE	<b>ACID</b> (Atomicity, Consistency, Isolation, and Durability) is a standard for RDBMS	<b>Base</b> (Basically Available, Soft state, Eventually Consistent) is a model of many NoSQL systems

# MongoDB

- A document-oriented database as opposed to a relational database
- Relational databases have fixed schemas
- Document-oriented databases have flexible schemas
- JSON-like documents

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)

## Relational database:

Name	School	Employer	Occupation
Lori	null	Self	Entrepreneur
Malia	Harvard	null	null

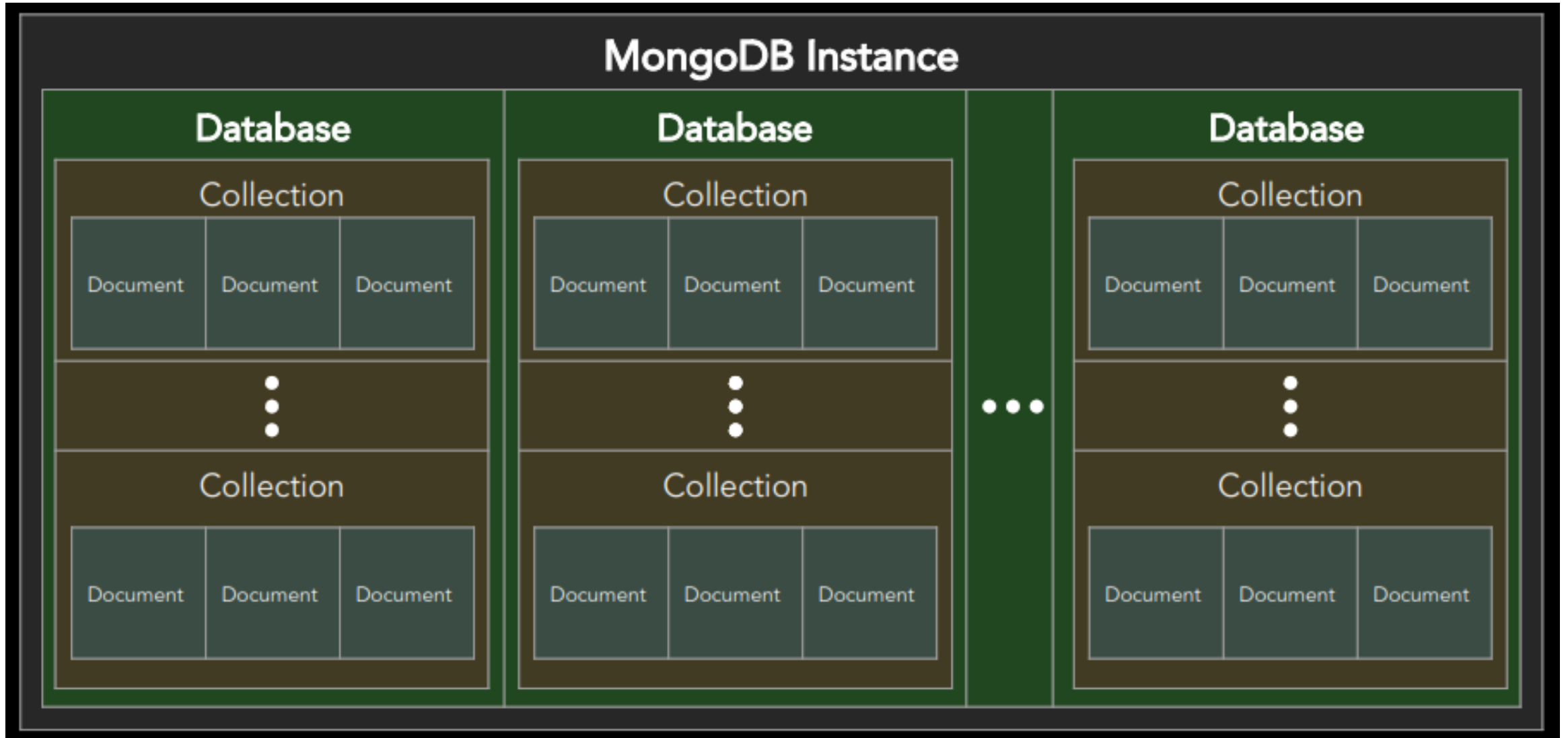
## Document-oriented DB:

```
{
  name: "Lori",
  employer: "Self",
  occupation: "Entrepreneur"
}
{
  name: "Malia",
  school: "Harvard"
}
```

# MongoDB Collections

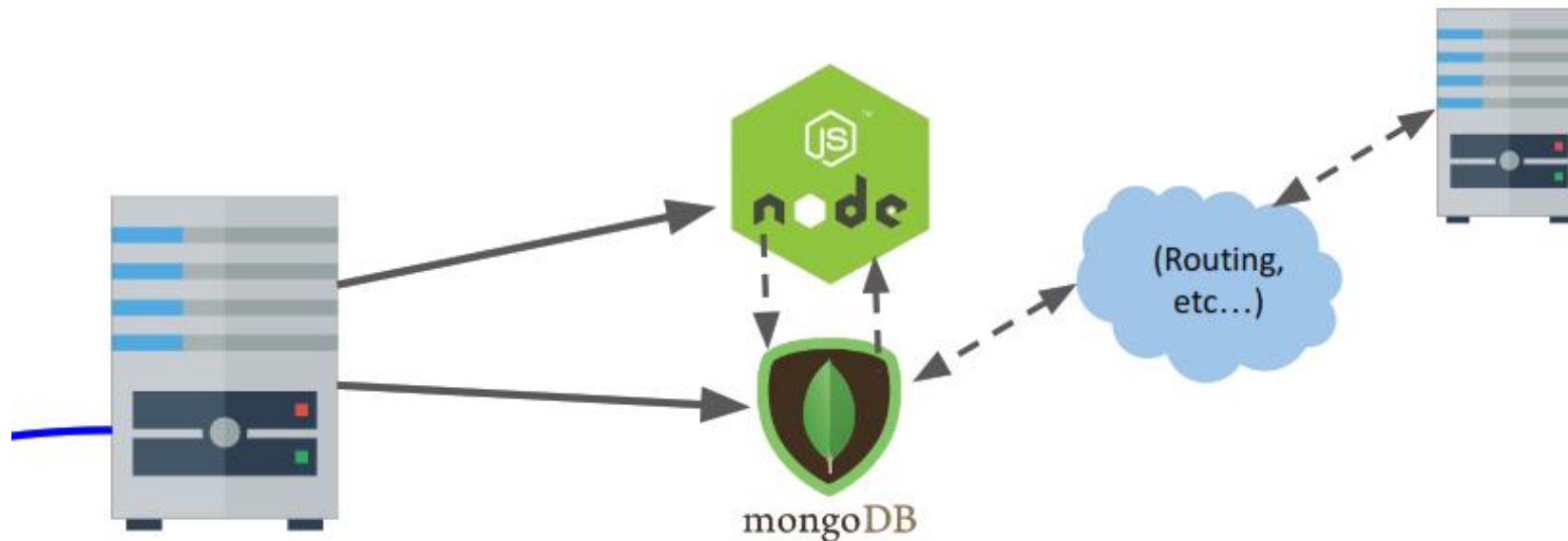
- Basic Storage is a collection
  - Set of json objects
  - Each has a unique identifier (generated by Mongo)
  - Fast lookup based on identifier

# Structure

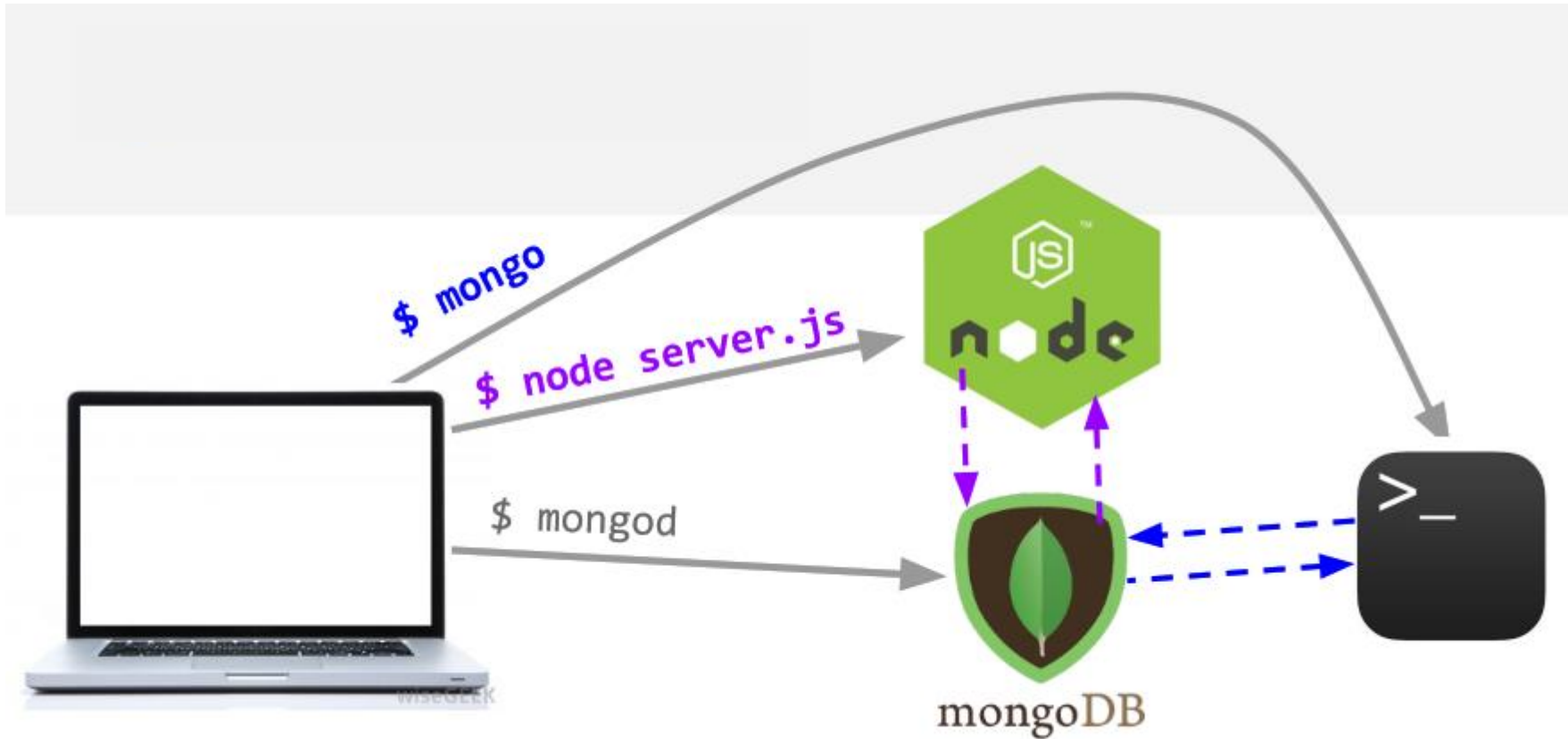


# MongoDB

- MongoDB is another software program running on the computer.
- It is also known as the MongoDB server.
- There are MongoDB libraries we can use in NodeJS to communicate with the MongoDB Server.
- The database the MongoDB Server manages might be local to the server computer or it could be stored on cloud storage.



# System overview



# Example

- Website has the following requirements.
  - Every post has the unique title, description and url.
  - Every post can have one or more tags.
  - Every post has the name of its publisher and total number of likes.
  - Every post has comments given by users along with their name, message, data-time and likes.
  - On each post, there can be zero or more comments.

# MongoDB schema

```
{
  _id: POST_ID
  title: TITLE_OF_POST,
  description: POST_DESCRIPTION,
  by: POST_BY,
  url: URL_OF_POST,
  tags: [TAG1, TAG2, TAG3],
  likes: TOTAL_LIKES,
  comments: [
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    },
    {
      user: 'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    }
  ]
}
```



# MongoDB CRUD

- `db.collection.insert()`
- `db.collection.find()`
- `db.collection.update()`
- `db.collection.delete()`