

Current shifts in tech

What is a database?

- A collection of data
- Anything can be stored in a database
- Types of databases:
 - Relational
 - NoSQL
 - Cloud
- A traditional database is relational
 - Utilizes a querying language - SQL

What is a Relational Database?

- A collection of **structured** data with **relationships** between them
- Organized using rows and columns
- SQL (Structured Querying Language)
- Common Relational Database Management Systems:
 - MySQL, OracleDB, SQL Server, etc.

Table: EECS 201 Assignments

AssignmentID	Topic	Basic Assignment Due date	Students Completed
1	Introduction and a dash of *nix	Fri, Feb 12	100
2	GITing started	Fri, Feb 19	78
3	Unix and You	Fri, Feb 26	112
4	Review and Regular Expressions	Wed, Mar 10	101
5	Git Gud	Fri, Mar 12	139
6	Text Editors	Wed, Mar 17	95

Also called the relation

Table: EECS 201 Assignments

AssignmentID	Topic	Basic Assignment Due date	Students Completed
1	Introduction and a dash of *nix	Fri, Feb 12	100
2	GITing started	Fri, Feb 19	78
3	Unix and You	Fri, Feb 26	112
4	Review and Regular Expressions	Wed, Mar 10	101
5	Git Gud	Fri, Mar 12	139
6	Text Editors	Wed, Mar 17	95

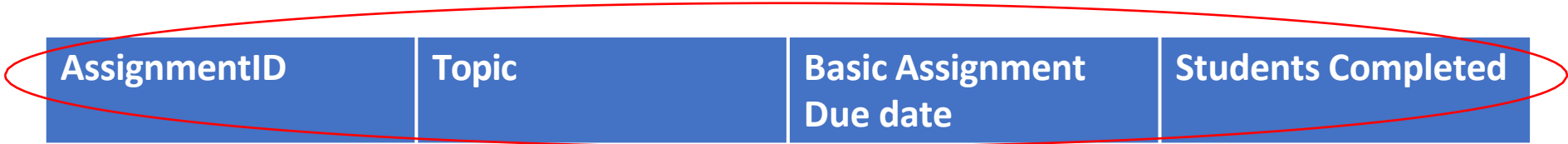
Table: EECS 201 Assignments

Primary key:
uniquely identifies each record in a table

AssignmentID	Topic	Basic Assignment Due date	Students Completed
1	Introduction and a dash of *nix	Fri, Feb 12	100
2	GITing started	Fri, Feb 19	78
3	Unix and You	Fri, Feb 26	112
4	Review and Regular Expressions	Wed, Mar 10	101
5	Git Gud	Fri, Mar 12	139
6	Text Editors	Wed, Mar 17	95

Table: EECS 201 Assignments

Columns/Attributes



AssignmentID	Topic	Basic Assignment Due date	Students Completed
1	Introduction and a dash of *nix	Fri, Feb 12	100
2	GITing started	Fri, Feb 19	78
3	Unix and You	Fri, Feb 26	112
4	Review and Regular Expressions	Wed, Mar 10	101
5	Git Gud	Fri, Mar 12	139
6	Text Editors	Wed, Mar 17	95

Table: EECS 201 Assignments

AssignmentID	Topic	Basic Assignment Due date	Students Completed
1	Introduction and a dash of *nix	Fri, Feb 12	100
2	GITing started	Fri, Feb 19	78
3	Unix and You	Fri, Feb 26	112
4	Review and Regular Expressions	Wed, Mar 10	101
5	Git Gud	Fri, Mar 12	139
6	Text Editors	Wed, Mar 17	95

Row/Tuple

What is MongoDB?

- MongoDB is a database
 - No-SQL (non-relational)
 - Nontraditional
 - Doesn't use rows or columns
 - Data is document based
 - Similar to JSON objects
 - BSON objects
 - Document is a datastructure



Document structure

- Structure:
 - **Data:** attribute-value pairs
 - **Objects:** separated by commas
 - “ ” holds data elements (data type list at end of slides)
 - [] holds arrays
 - { } holds documents- can have embedded documents
 - **_id field:** always first in a document; unique & it can be any datatype except an Array

Example

```
var EECS = {  
  {  
    _id: ObjectId("5099803df3f4948bd2f98391"), _id_field  
    instructor:"true",  
    name:{ firstName:"Brandon", lastName:"Nguyen" }  
    classes:["EECS_201"]  
    instructionalAides:[  
      { firstName: "Sowgandhi",  
        major: "CSE"  
      },  
      { firstName: "Arav",  
        major: "dSE"  
      }  
    ]  
  },  
  {  
    _id: ObjectId("5099803df3f4948bd2f98392"),  
    instructor:"true",  
    name:{ firstName:"Prof2", lastName:"Last" }  
    classes:["EECS_281", "EECS_183"]  
    instructionalAides:[  
      { firstName: "IA",  
        major: "CSE"  
      }  
    ]  
  }  
}
```

Data example

Example

```
var EECS = {
  {
    _id: ObjectId("5099803df3f4948bd2f98391"),
    instructor:"true",
    name:{ firstName:"Brandon", lastName:"Nguyen" }
    classes:["EECS_201"]
    instructionalAides:[
      { firstName: "Sowgandhi",
        major: "CSE"
      },
      { firstName: "Arav",
        major: "dSE"
      }
    ]
  },
  {
    _id: ObjectId("5099803df3f4948bd2f98392"),
    instructor:"true",
    name:{ firstName:"Prof2", lastName:"Last" }
    classes:["EECS_281", "EECS_183"]
    instructionalAides:[
      { firstName: "IA",
        major: "CSE"
      }
    ]
  }
}
```

Data element

Example

```
var EECS = {
  {
    _id: ObjectId("5099803df3f4948bd2f98391"),
    instructor:"true",
    name:{ firstName:"Brandon", lastName:"Nguyen" },
    classes:["EECS_201"],
    instructionalAides:[
      { firstName: "Sowgandhi",
        major: "CSE"
      },
      { firstName: "Arav",
        major: "dSE"
      }
    ]
  },
  {
    _id: ObjectId("5099803df3f4948bd2f98392"),
    instructor:"true",
    name:{ firstName:"Prof2", lastName:"Last" },
    classes:["EECS_281", "EECS_183"],
    instructionalAides:[
      { firstName: "IA",
        major: "CSE"
      }
    ]
  }
}
```

Array

Example

```
var EECS = {
  {
    _id: ObjectId("5099803df3f4948bd2f98391"),
    instructor:"true",
    name:{ firstName:"Brandon", lastName:"Nguyen" },
    classes:["EECS_201"],
    instructionalAides:[
      { firstName: "Sowgandhi",
        major: "CSE"
      },
      { firstName: "Arav",
        major: "dSE"
      }
    ]
  },
  {
    _id: ObjectId("5099803df3f4948bd2f98392"),
    instructor:"true",
    name:{ firstName:"Prof2", lastName:"Last" },
    classes:["EECS_281", "EECS_183"],
    instructionalAides:[
      { firstName: "IA",
        major: "CSE"
      }
    ]
  }
}
```

Document

Running Queries using Mongo Shell

- `db.EECS.insert(...)`
- `db.EECS.find(...)`
 - Can query by embedded documents
 - Ex) `db.EECS.find({ name:{ firstName:"Brandon", lastName:"Nguyen" } })`
 - Must be an exact match to return the document
 - Can query by nested fields
 - Ex) `db.EECS.find({ "name.firstName" : "Brandon" })`
 - Will return any document with a name, firstName Brandon
 - Can query using arrays
 - Multiple ways: by matching an entire array, by elements in array, matching some criteria

Example

```
var EECS = {
  {
    _id: ObjectId("5099803df3f4948bd2f98391"),
    instructor:"true",
    name:{ firstName:"Brandon", lastName:"Nguyen" }
    classes:["EECS_201"]
    instructionalAides:[
      { firstName: "Sowgandhi",
        major: "CSE"
      },
      { firstName: "Arav",
        major: "dSE"
      }
    ]
  },
  {
    _id: ObjectId("5099803df3f4948bd2f98392"),
    instructor:"true",
    name:{ firstName:"Prof2", lastName:"Last" }
    classes:["EECS_281", "EECS_183"]
    instructionalAides:[
      { firstName: "IA",
        major: "CSE"
      }
    ]
  }
}
```

db.EECS.find({ name:{
 firstName:"Brandon",
 lastName:"Nguyen" } })

Running Queries using Mongo Shell

- `db.EECS.insert(...)`
- `db.EECS.find(...)`
 - Can query by embedded documents
 - Ex) `db.EECS.find({ name:{ firstName:"Brandon", lastName:"Nguyen" } })`
 - Must be an exact match to return the document
 - Can query by nested fields
 - Ex) `db.EECS.find({ "name.firstName" : "Brandon" })`
 - Will return any document with a name, firstName Brandon
 - Can query using arrays
 - Multiple ways: by matching an entire array, by elements in array, matching some criteria

Why Mongo?

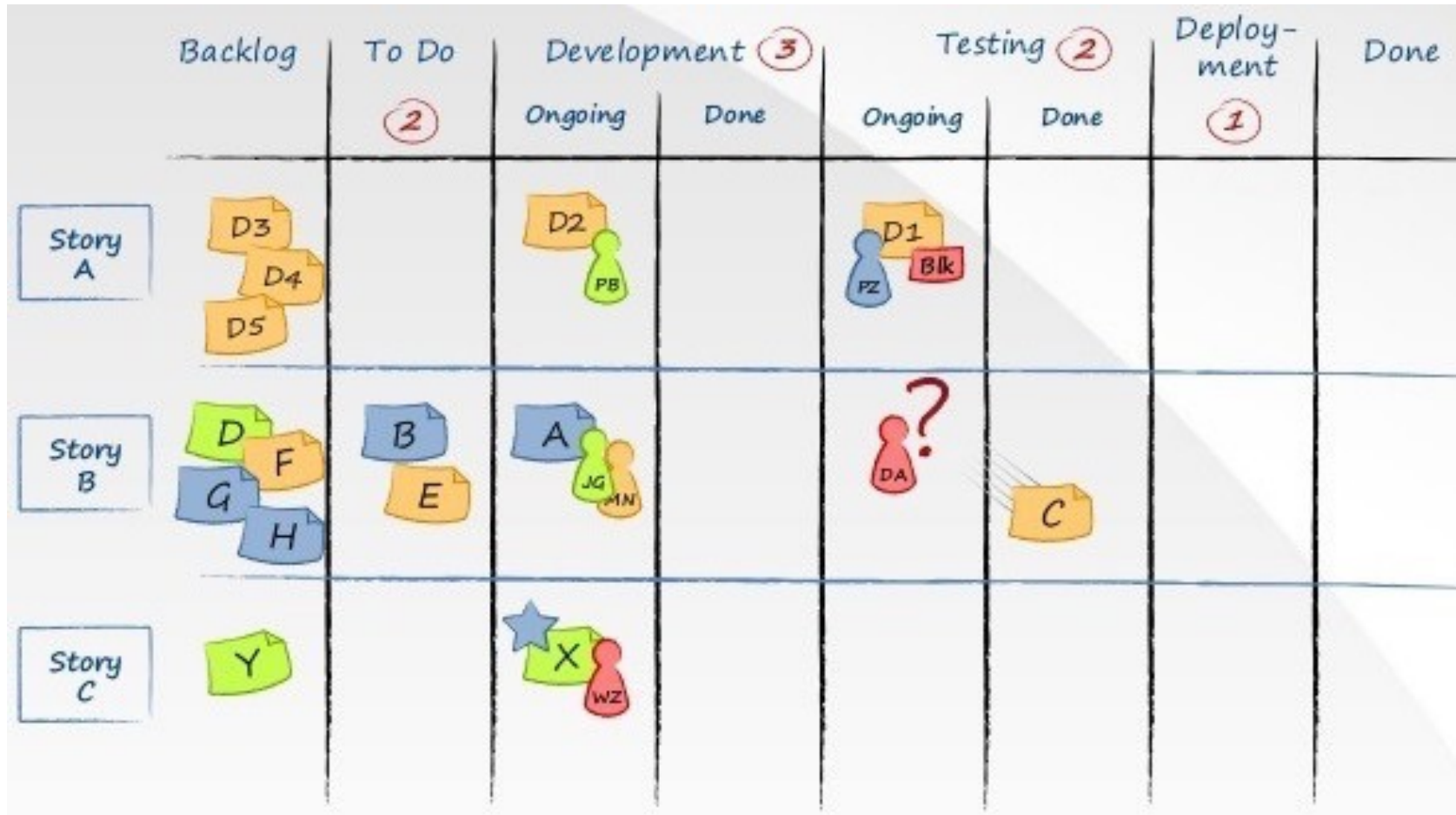
- Limitations of SQL Databases
 - Requires all data to fit into a table
 - Requires a lot of setup
 - users must scale the data using vertically scalable servers
- Mongo was built off these limitations
 - You can enter data any way you want
 - Cheap and open source
 - Maintenance is a lot cheaper – its simple to use
 - Horizontally scalable (can use cheaper servers)

What is Agile ?

- Software development process/ technical philosophy
 - Rapid delivery of working software
 - Cross-functional teamwork
 - customer needs/company goals
 - Responding to change
- Agile Manifesto- published in 2001 further describes the 12 principles
 - <https://agilemanifesto.org/>

Common frameworks

- Scrum
 - Goal: deliver highest value to stakeholders
 - Focuses on reflection
- Kanban
 - Implements Agile and DevOps
 - DevOps= process of automating processes between software development and IT
 - Goal: transparency between team members
- Often used together- “Scrumban”



Agile for developers

- Focus on sustainable development
 - Managing code- minimize bugs
 - Automated testing
 - Continuous deployment
- Quality > Scope or Schedule
- Work alongside PM, design, QA and operations
- Git
 - Tasks are essentially Git branches
 - Helps with distributed testing
 - Idea of continuous integration

Why Agile?

- No project failure
- Disciplined program management
- Easy to adapt to change
- Useful for app and software startups get going

Thank you!

MongoDB datatypes

- **String** – This is the most commonly used datatype to store data. String in MongoDB must be UTF-8 valid.
- **Integer** – This type is used to store a numerical value. Integer can be 32 bit or 64 bit depending upon your server.
- **Boolean** – This type is used to store a boolean (true/ false) value.
- **Double** – This type is used to store floating point values.
- **Min/ Max keys** – This type is used to compare a value against the lowest and highest BSON elements.
- **Arrays** – This type is used to store arrays or list or multiple values into one key.
- **Timestamp** – This can be handy for recording when a document has been modified or added.
- **Object** – This datatype is used for embedded documents.
- **Null** – This type is used to store a Null value.
- **Symbol** – This datatype is used identically to a string; however, it's generally reserved for languages that use a specific symbol type.
- **Date** – This datatype is used to store the current date or time in UNIX time format. You can specify your own date time by creating object of Date and passing day, month, year into it.
- **Object ID** – This datatype is used to store the document's ID.
- **Binary data** – This datatype is used to store binary data.
- **Code** – This datatype is used to store JavaScript code into the document.
- **Regular expression** – This datatype is used to store regular expression.

https://www.tutorialspoint.com/mongodb/mongodb_datatype.htm