

CMSC335

Web Application Development with JavaScript



MongoDB

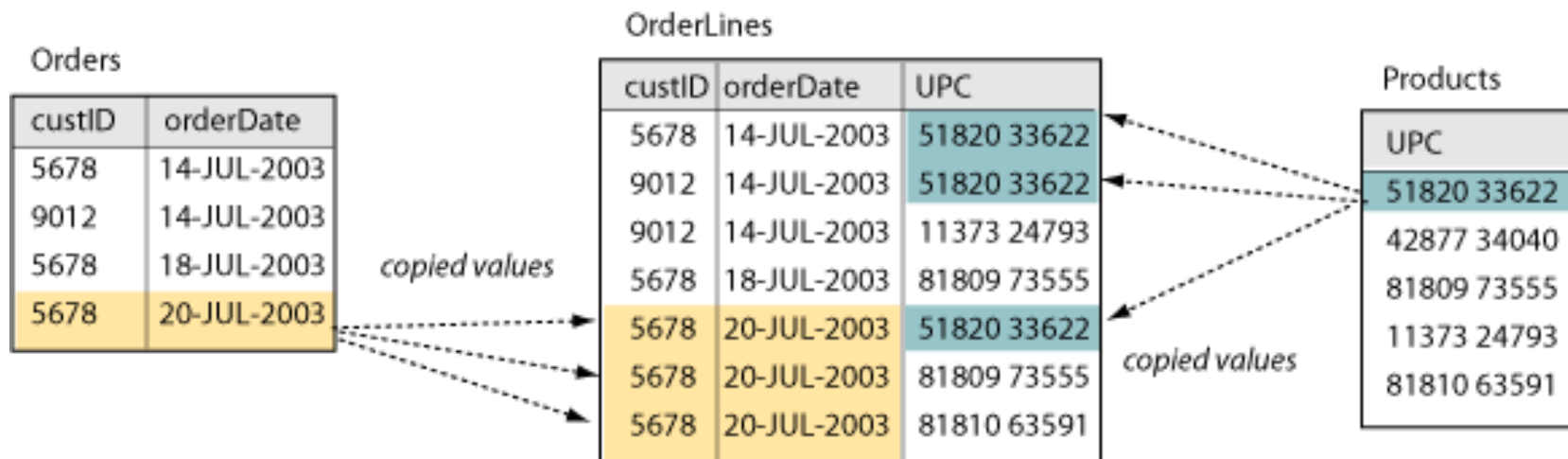
Department of Computer Science

University of MD, College Park

Slides material developed by Ilchul Yoon, Nelson Padua-Perez

Databases

- An organized collection of data/information
- Relational database (SQL based)(e.g., MySQL)
 - Store data as tables
 - **Structured**, low redundancy of information across tables



Source: <https://web.csulb.edu/colleges/coe/cecs/dbdesign/dbdesign.php?page=manymany.php>

Non-relational database

- Called NoSQL databases
 - Examples: MongoDB, Cassandra, Redis, ...
- Less structured (or... may say **flexible** data structure)

```
{  
  name: "Pikachu",  
  imageUrl: "https://assets.pokemon.com/assets25.png",  
  stats: {  
    hp: 3,  
    attack: 5,  
    defense: 3,  
    speed: 6  
  }  
}
```



- Document database
- A record in mongoDB is a document, which is:
 - a data structure composed of field and value pairs

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

- **Stores documents in “Collections”**
 - Similar to tables in relational databases
- We can install mongoDB locally but will use the Cloud version (mongoDB Atlas)
- See: <https://www.cs.umd.edu/~nelson/classes/resources/web/mongodb/>

CRUD operations on Resources

- CRUD - Create, Read, Update, Destroy (or Delete)
- Typical examples of resources:
 - Data stored in a database
 - Pages
 - Files
- **Examples:** DatabaseExamples
 - Check README.txt. Assuming Mongo account has been set and credentialsDontPost/.env has been updated with your credentials, execute in Node.js
 - listDatabase.js, insertMovies.js, listAllMovies.js, lookUpMovies.js, updateMovie.js deleteMovies.js, clearCollection.js