















ACLS


Graph-based



SS 2020 – Week 12
May 4





Schedule





Calendar Week	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Module Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Date	17.02.	24.02.	02.03.	09.03.	16.03.	23.03.	30.03.	06.04.	13.04.	20.04.	27.04.	04.05.	11.05.	18.05.
Topic														



15.06.2020
10.00am

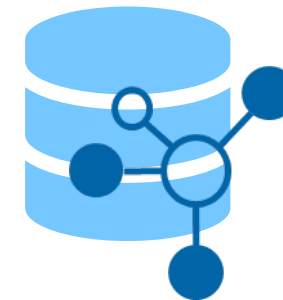
February
March
April
May

 Introduction & RDB
 Structured Query Language
 Database & Python
 Database & R

 Data Warehouse
 Not only SQL
 Graph Database
 Special

Content

• Review		5'
• Graph-based	🎓	20'
• Relational → LPG	🎓	20'
• Neo4j and Cypher	🎓	45'
• Exercise	🔧	135'



Content

• Review		5'
• Graph Database	🎓	20'
• Relational → LPG	🎓	20'
• Neo4j and Cypher	🎓	45'
• Exercise	🔧	135'



Review (1/3)

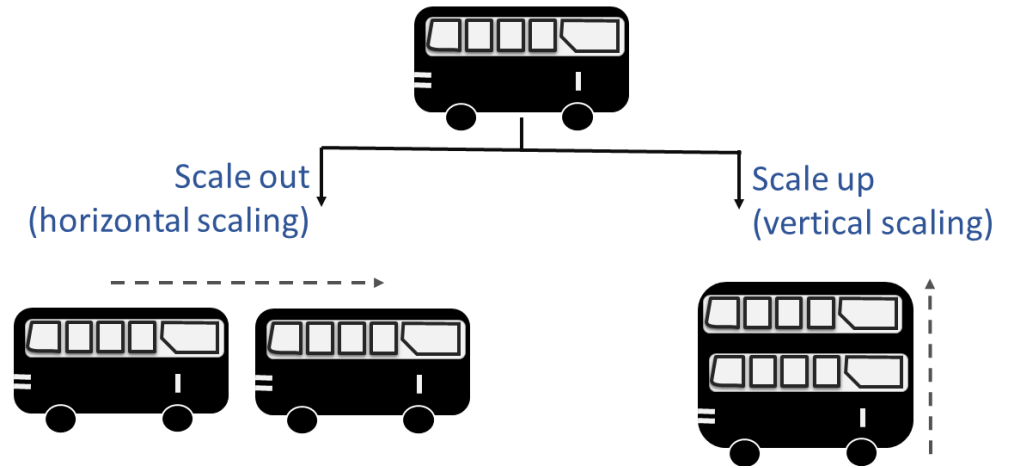
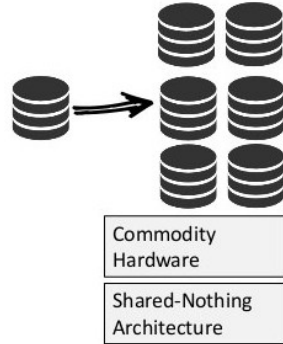
NoSQL – Horizontal Scalability

Scale-up vs Scale-out

Scale-Up (*vertical scaling*):



Scale-Out (*horizontal scaling*):



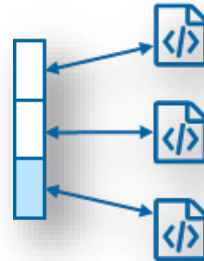
Review (2/3)

NoSQL - Types

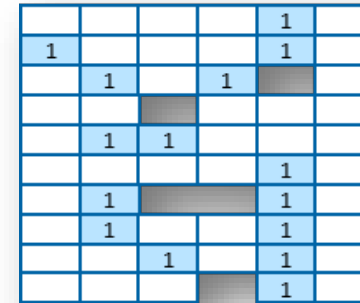
Key-Value



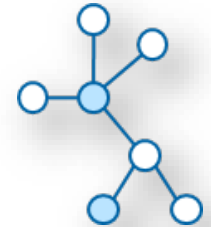
Document



Column



Graph

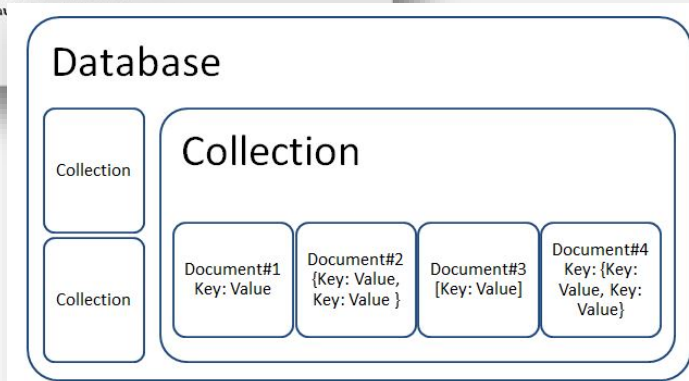



Review (3/3)

Document-oriented

Customer Document

```
"customer" =
{
  "id": "Customer:1",
  "firstName": "John",
  "lastName": "Wick",
  "age": 25,
  "address": {
    "country": "US",
    "city": "New York",
    "state": "NY",
    "street": "21 2nd Street",
  },
  "hobbies": [ Football, Hiking ],
  "phoneNumbers": [
    {
      "type": "Home",
      "number": "212 555-1234"
    },
    {
      "type": "Office",
      "number": "212 555-1234"
    }
  ]
}
```

```
# import mongoDB module
import pymongo

# create a new connection to the server
myConn = pymongo.MongoClient('server url')

# select a specific database on the server
database = myConn['databasename']

# select a collection in the database
collection = database['collectionname']

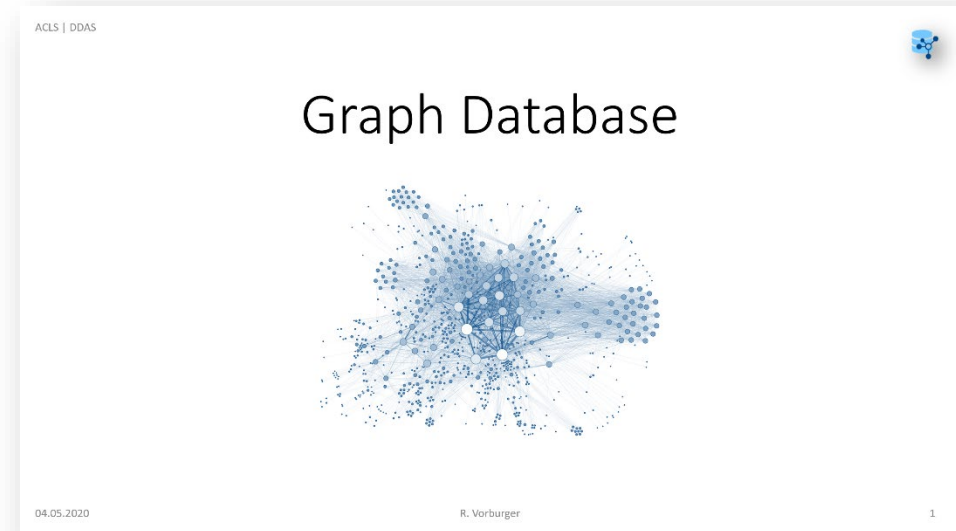
# create new document
new_doc = {'key1':value1, 'key2':value2, ... , 'keyN':valueN}

# insert new document
collection.insert_one(new_doc)

# retrieve documents with a given content
documents = collection.find({'some_key':'some_value'})
```

Content

• Review		5'
• Graph Database	🎓	20'
• Relational → LPG	🎓	20'
• Neo4j and Cypher	🎓	45'
• Exercise	🔧	135'



Content

• Review		5'
• Graph Database	🎓	20'
• Relational → LPG	🎓	20'
• Neo4j and Cypher	🎓	45'
• Exercise	🔧	135'

ACLS | DDAS

Relational → LPG

04.05.2020

R. Vorburger


1

Content

• Review		5'
• Graph Database	🎓	20'
• Relational → LPG	🎓	20'
• Neo4j and Cypher	🎓	45'
• Exercise	🔧	135'





ACLS | DDAS

Neo4j & Cypher



04.05.2020 R. Vorburger 1

Content

• Review		5'
• Graph Database		20'
• Relational → LPG		20'
• Neo4j and Cypher		45'
• Exercise		135'

MSc ACLS Databases and Data Architecture Systems SS20

[Dashboard](#) / [My courses](#) / [ACLS DDAS SS20](#) / [Exercises](#) / [Week 12](#)

Week 12

 [graph_based.pdf](#)

[Download folder](#)