Large-Scale and Multi-Structured Databases

Introduction to the Course

Academic Year 2024-2025

Prof Pietro Ducange

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

IN SUPREMÆ DIGNITATIS 1343

UNIVERSITÀ DI PISA

CROSSLAB
Innovation for industry 4.0

Who is Talking to You?

Pietro Ducange

- Born in Apulia, South of Italy
- Master Degree in Computer Engineering in 2005, University of Pisa
- PhD in Information Engineering in 2009, University of Pisa
- Post-doc Researcher 2009-2014, University of Pisa
- Associate Professor 2014-2019, eCampus University
- Associate Professor 2019-on going, University of Pisa

Pietro's Research Activity

Main Research Topic:

- Big Data Mining and Analytics
- Text Analysis
- Explainable Artificial Intelligence

Member of:

AI&RD Research Group @ DDI

https://ai.dii.unipi.it

Cloud Computing, Big Data and Cyber Security Lab@DII:

https://crosslab.dii.unipi.it/cloud-computing-big-data-cybersecurity-lab

Publication Records:

https://scholar.google.it/citations?user=HCgZqXEAAAAJ&hl=it

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

IN SUPREMÆ DIGNITATIS 1343

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

The Course

Large Scale and Multi-Structured Databases

9 CFU-> 90 Hours

Program Degrees:

- M.Sc. in Artificial Intelligence and Data Engineering (1-2 Year)
- M.Sc. in Computer Engineering (1 Year)

Syllabus

Introduction and Motivations: Introduction to the Course, The Big Data Era, The Database Revolutions

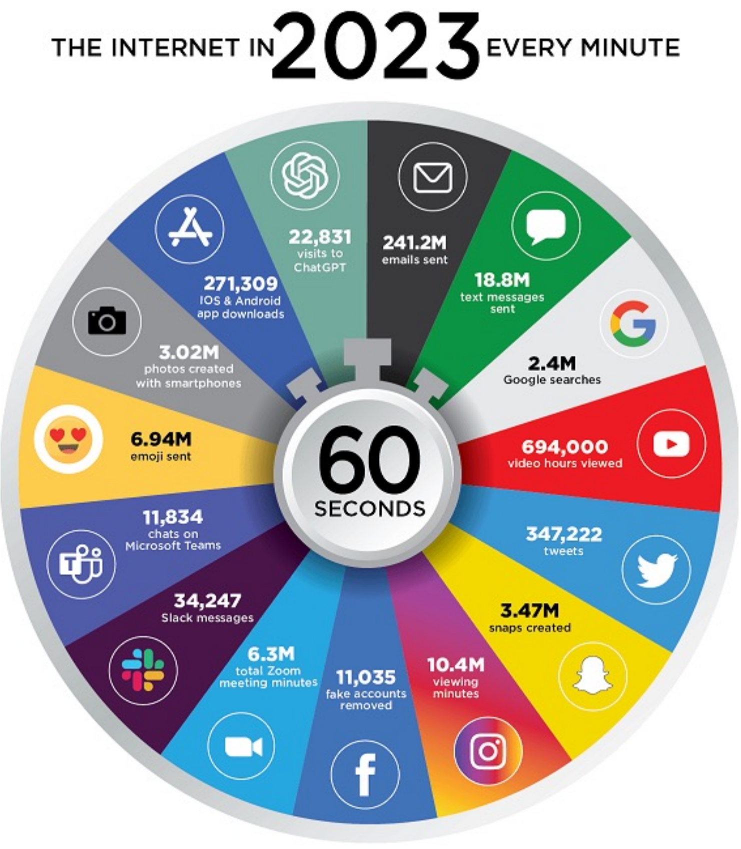
Fundamentals and properties of the NoSQL databases: ACID vs BASE properties, The Cap Theorem, Scalability, Sharding, Replication, Consistency

Architectures of NOSQL databases: Document Databases, Key-values Databases, Column Databases, Graph Databases

Recaps: Recap of Software Engineering and Java (basics, connections and queries towards SQL databases).

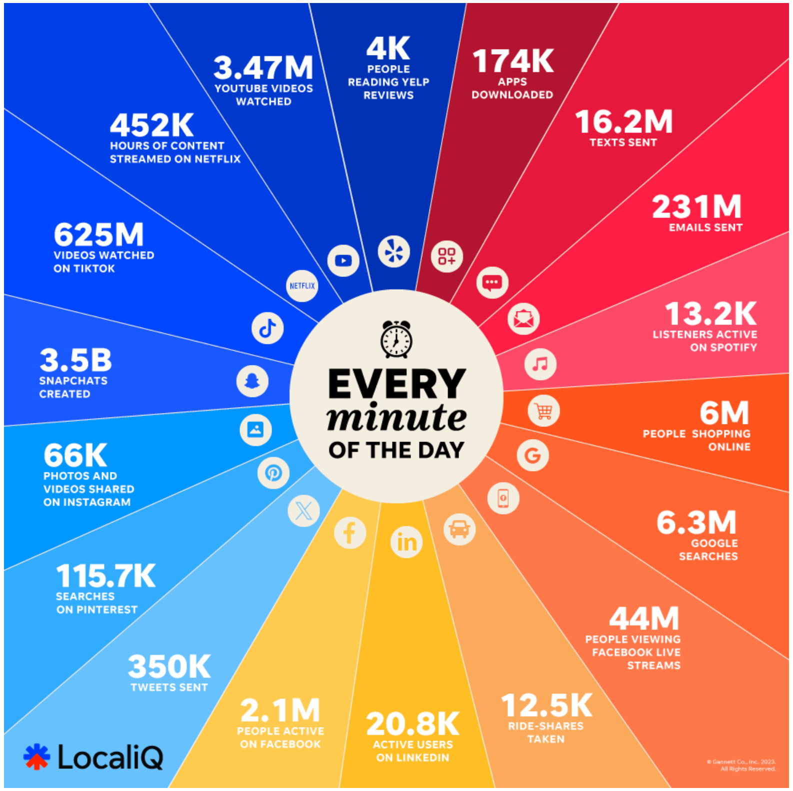
Modern Infrastructures for NoSQL Databases: REDIS, MongoDB, Neo4J (Installation, configuration, CRUD operations, main queries)

The Big Data Era



Created by: eDiscovery Today & LTMG

2024



The Data Base Revolutions

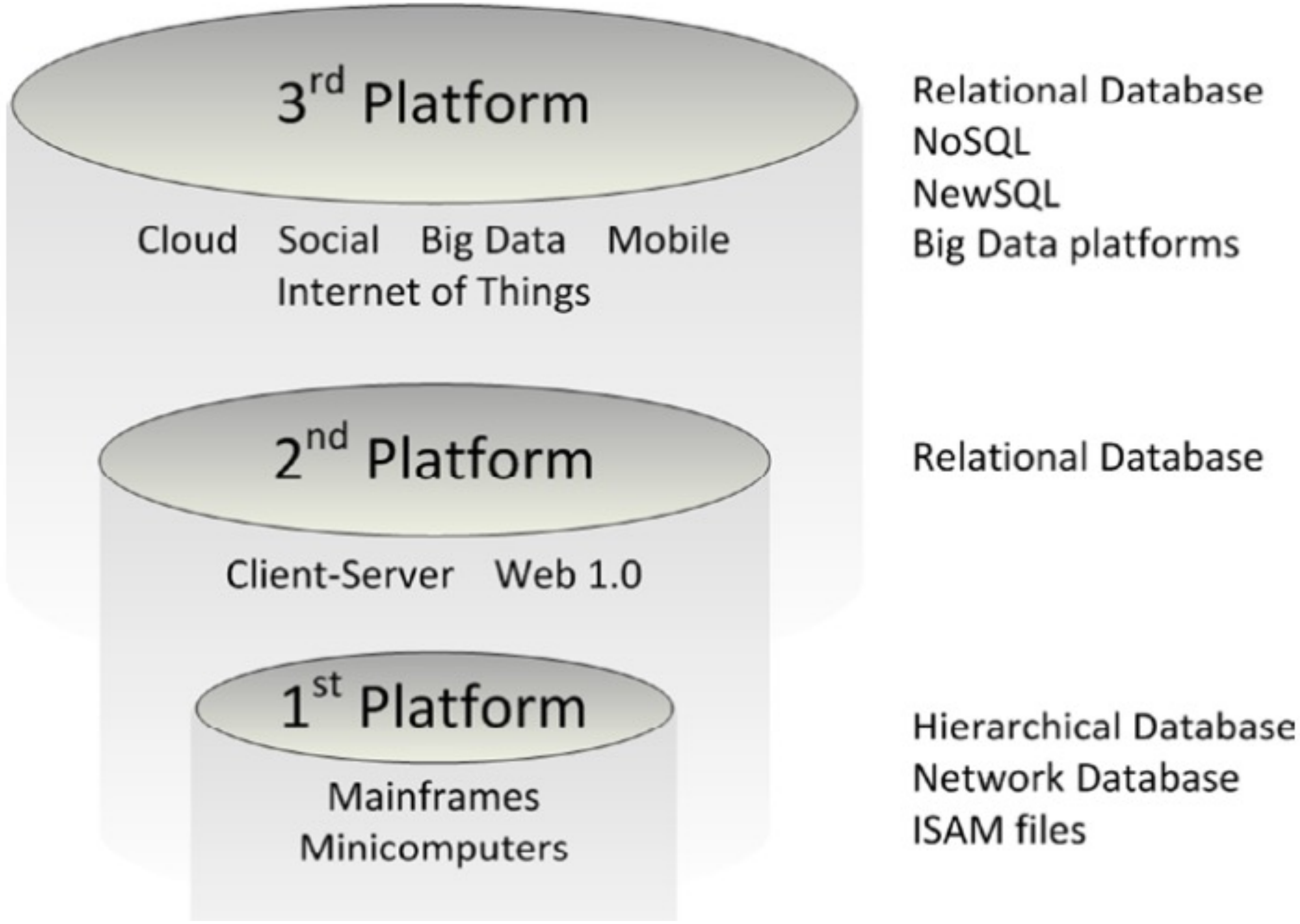
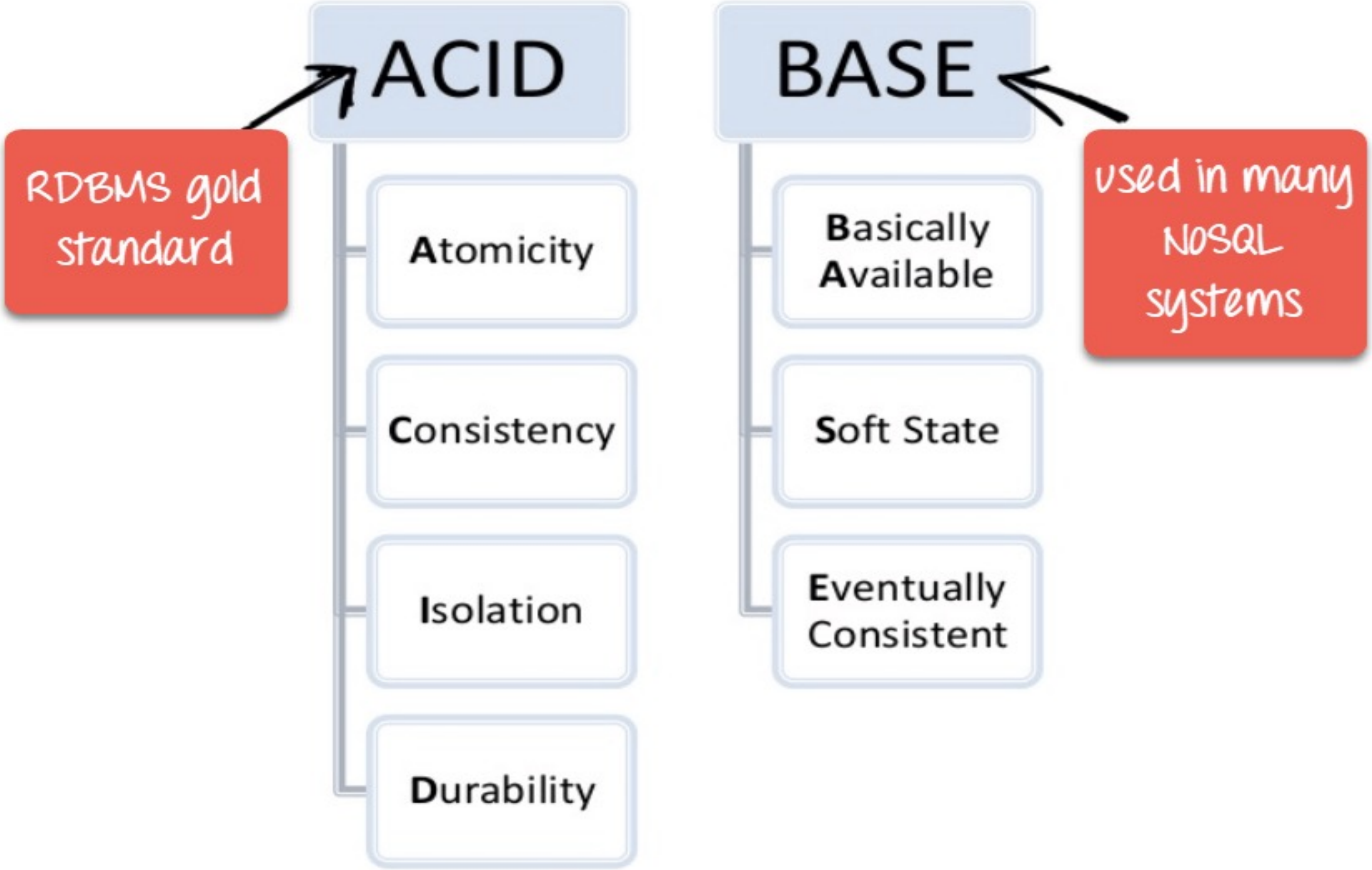


Image extracted from "Guy Harrison, Next Generation Databases, Apress, 2015"

ACID vs BASE



RDBMS gold standard

ACID

- Atomicity
- Consistency
- Isolation
- Durability

used in many NoSQL systems

BASE

- Basically Available
- Soft State
- Eventually Consistent

Image extracted from: https://www.guru99.com/sql-vs-nosql.html

Key-Value Databases

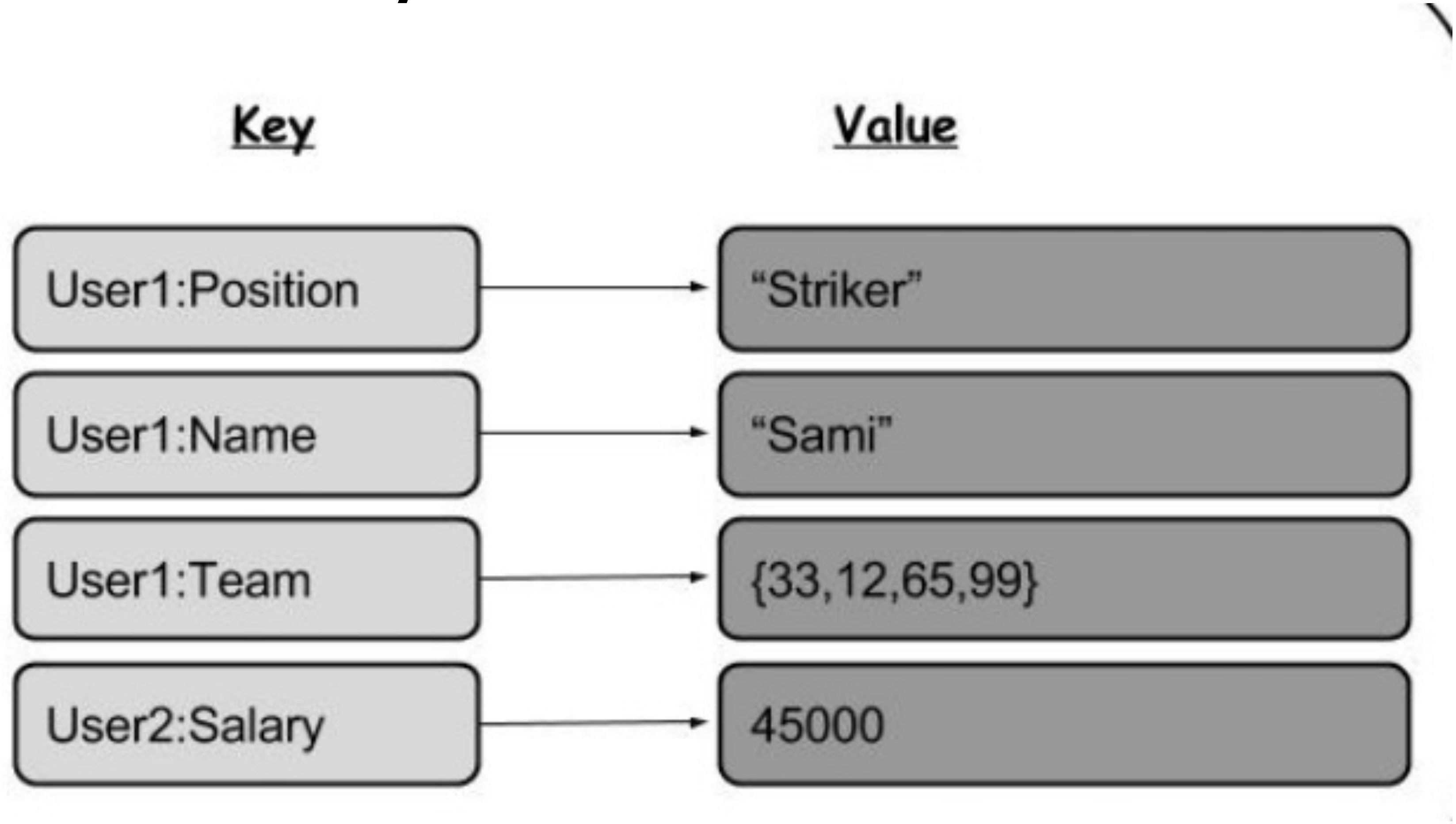
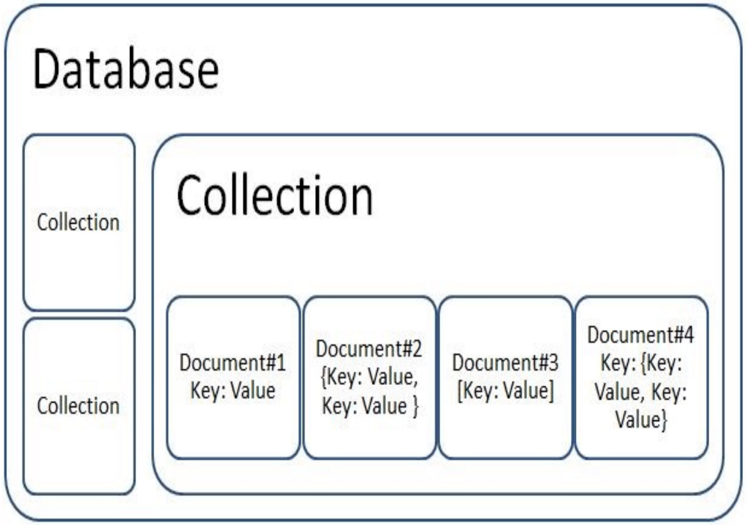
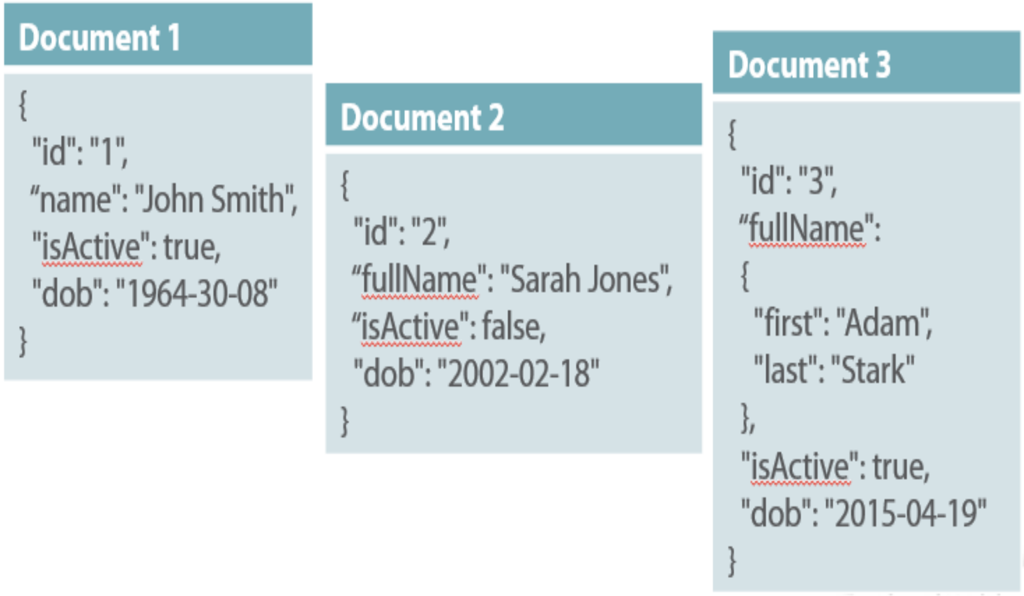


*Image extracted from:*https://www.researchgate.net/figure/Key-value-NoSQL-Database_fig1_332188615

Document Databases





Document 1

```
{
  "id": "1",
  "name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

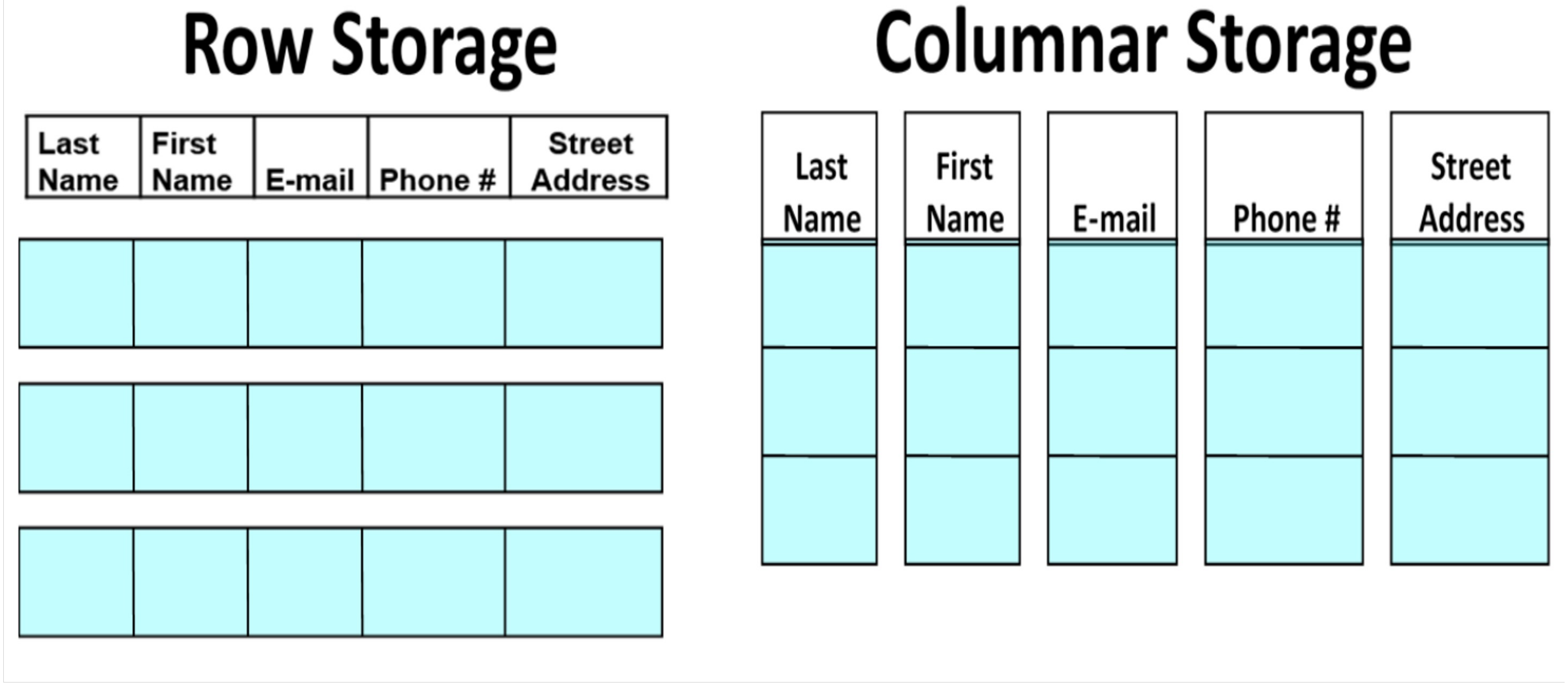
```
{
  "id": "2",
  "fullName": "Sarah Jones",
  "isActive": false,
  "dob": "2002-02-18"
}
```

Document 3

```
{
  "id": "3",
  "fullName":
  {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```

Images extracted from: https://dzone.com/articles/a-primer-on-open-source-nosql-databases
https://lennilobel.wordpress.com/2015/06/01/relational-databases-vs-nosql-document-databases/

Column Databases



Graph Databases

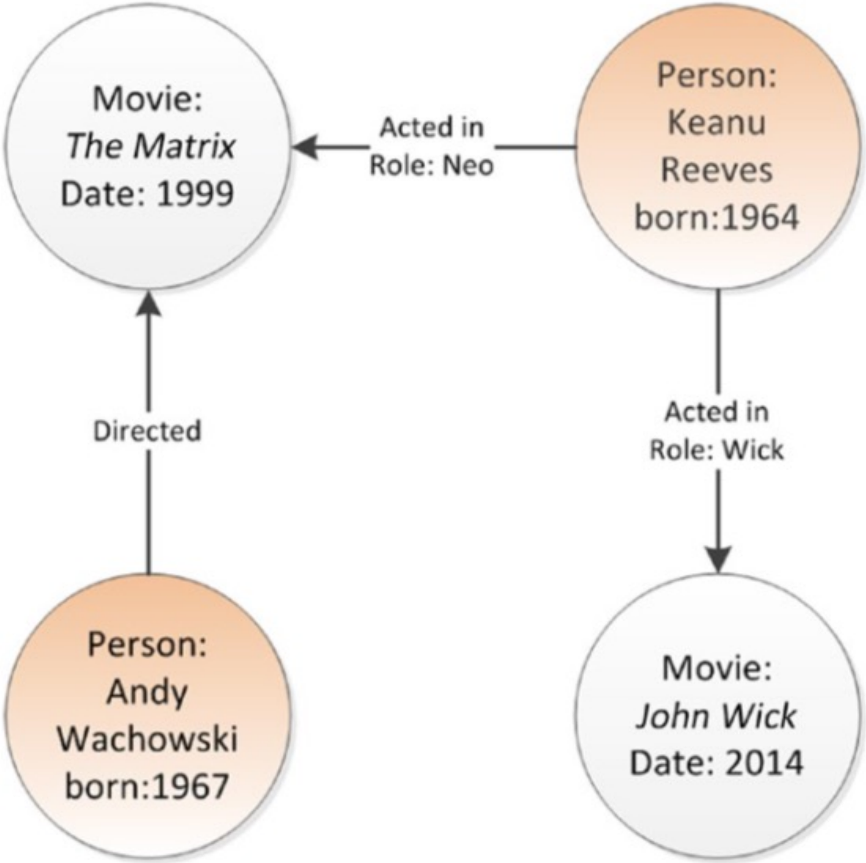


Image extracted from "Guy Harrison, Next Generation Databases, Apress, 2015"

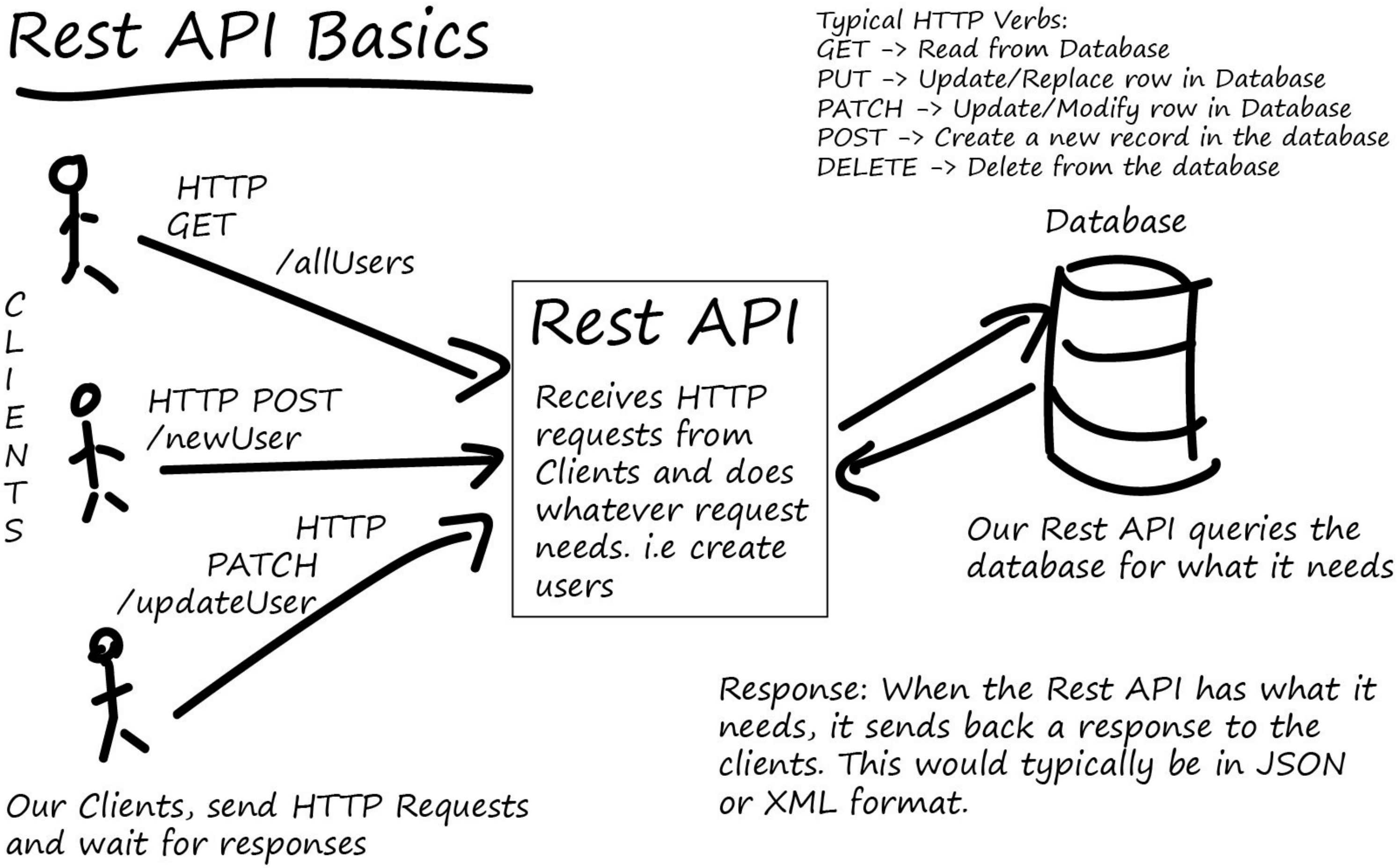
Recaps Software Engineering

- Functional and non-functional requirements
- Use cases definitions
- UML Diagrams
- Some exercises on designing a complete application

Recaps on Java (4 hours)

- Hello World! In Java using an IDE (Eclipse/IntelliJ)
- Some Java Programming Exercises
- Connection to MySQL server: JDBC (using Maven for handling dependencies)
- Simple application connecting a relational DB which export its functionalities using RESTful API.

Recaps on Java (4 hours)



Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database

Receives HTTP requests from Clients and does whatever request needs. i.e create users

Our Rest API queries the database for what it needs

Our Clients, send HTTP Requests and wait for responses

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

Modern NoSQL Infrastructures

Type	Example
Key-Value Store	redis riak
Wide Column Store	HBASE cassandra
Document Store	mongoDB CouchDB relax
Graph Store	Neo4j InfiniteGraph The Distributed Graph Database

amazon DynamoDB

Learning Outcomes: *Knowledge*

At the end of the course:

- The student will have acquired knowledge about ***methodologies*** and ***tools*** and for the design of ***non-relational databases***.

- The student will acquire knowledge about the ***architectures***, ***performances*** and ***costs*** of modern infrastructures for the management of complex data.

- The student will be able to correctly ***set up*** a ***project*** for the management of multi-structured and large data, integrating it into a ***real computer application*** and choosing in an appropriate manner the design and implementation strategies.

Assessment Criteria of Knowledge

Group activities will be proposed to assess theoretical and practical knowledge.

Group activities will be proposed to the ***working groups*** with the objective of:

- ***deepening*** of theoretical and technical issues
- ***implementing*** of technical projects

Periodic classroom discussions between the teacher and the group of students developing the above activities will be organized.

Skills

At the end of the course the student will be able to:

- ***Design*** a non-relational database based on the ***requirements*** (functional and non-functional) of a specific ***application***.

- Use modern ***technological infrastructures*** for the management of non-relational databases (Redis, MongoDB, Neo4j)

Assessment Criteria of Skills

During lab class:

- The student will be shown how to ***install*** and ***configure*** some of the modern technological infrastructures for the management of non-relational databases.

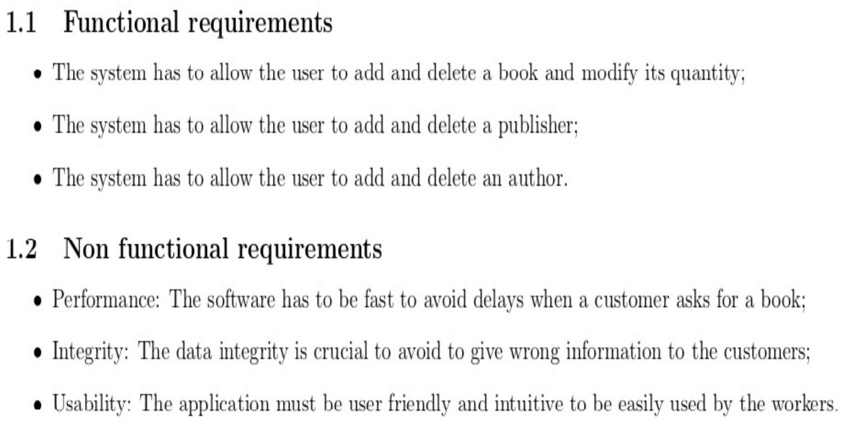
- ***Practical activities*** will be proposed for the creation, management and querying of different non-relational databases.

- Group activities will be proposed for the ***in-depth study*** of technical issues and for the implementation of educational projects.

Prerequisites

- Programming in ***JAVA*** (including the use of an IDE and RESTful API)-> Skills acquired in ***Programmazione Avanzata***
- Design and query of ***relational databases***
- Basics of ***Software Engineering*** (including realization of UML Diagrams)
- Basics of ***Unix-Based*** Operating Systems

About Software Engineering Skills

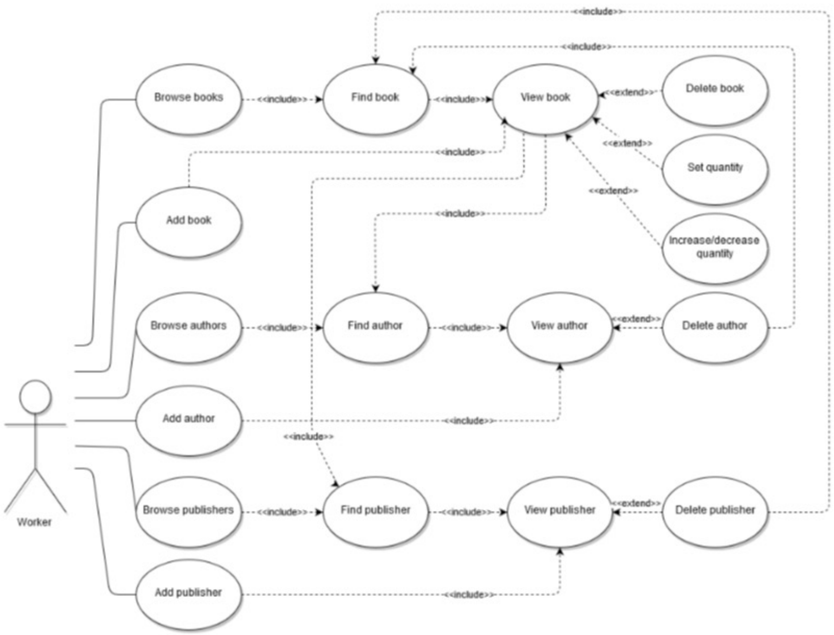


1.1 Functional requirements

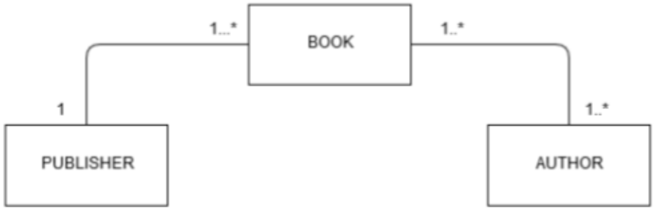
- The system has to allow the user to add and delete a book and modify its quantity;
- The system has to allow the user to add and delete a publisher;
- The system has to allow the user to add and delete an author.

1.2 Non functional requirements

- Performance: The software has to be fast to avoid delays when a customer asks for a book;
- Integrity: The data integrity is crucial to avoid to give wrong information to the customers;
- Usability: The application must be user friendly and intuitive to be easily used by the workers.

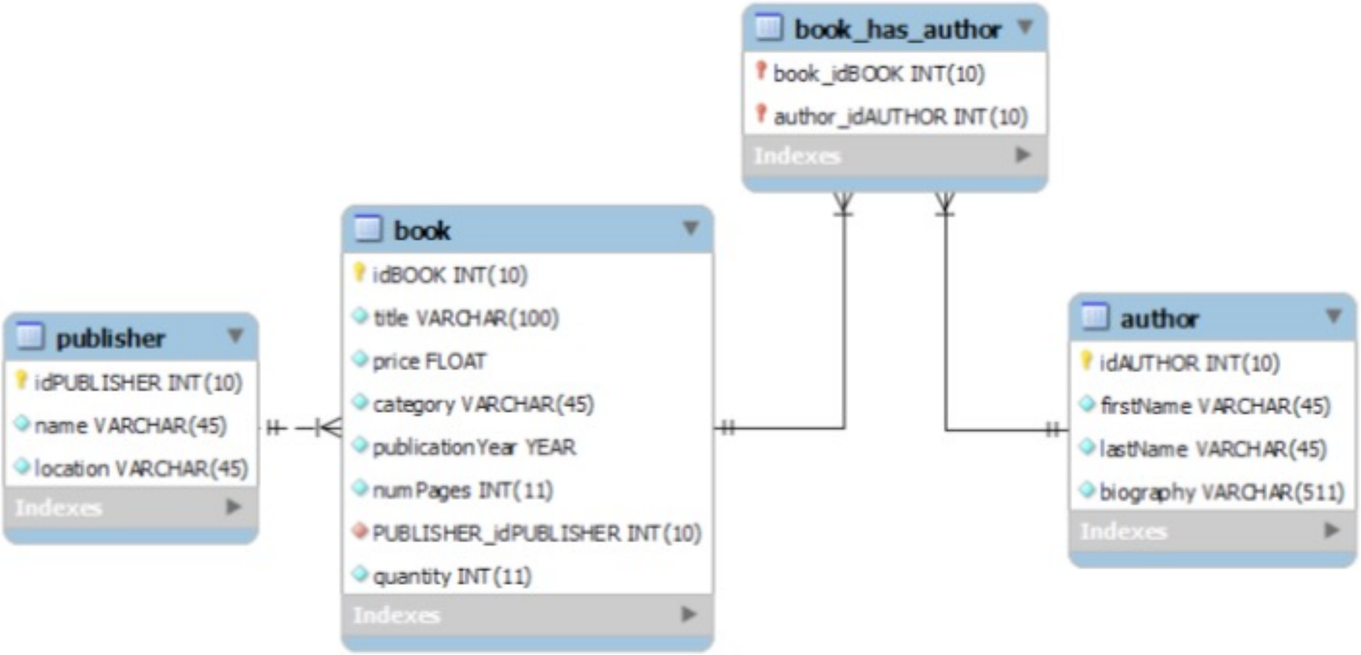


UML Use Case Diagrams



UML Analysis Class Diagrams

About Relational Databases



book_has_author
- book_idBOOK INT(10)
- author_idAUTHOR INT(10)
- Indexes

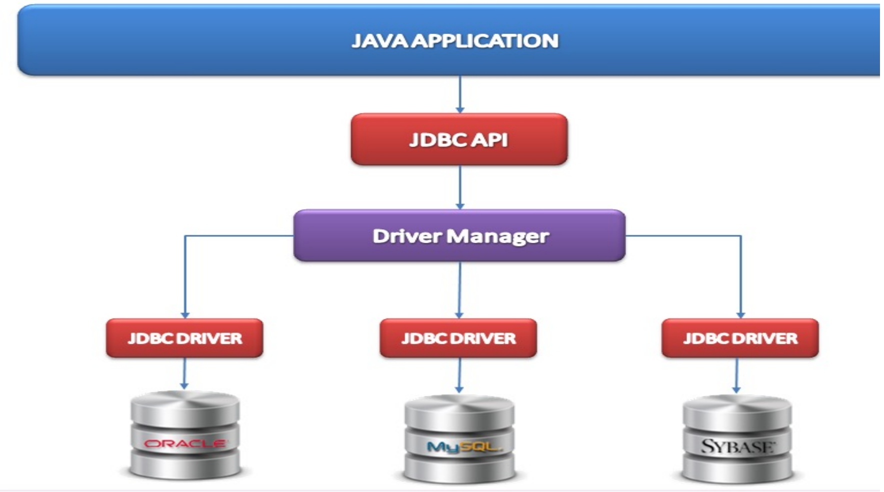
book
- idBOOK INT(10)
- title VARCHAR(100)
- price FLOAT
- category VARCHAR(45)
- publicationYear YEAR
- numPages INT(11)
- PUBLISHER_idPUBLISHER INT(10)
- quantity INT(11)
- Indexes

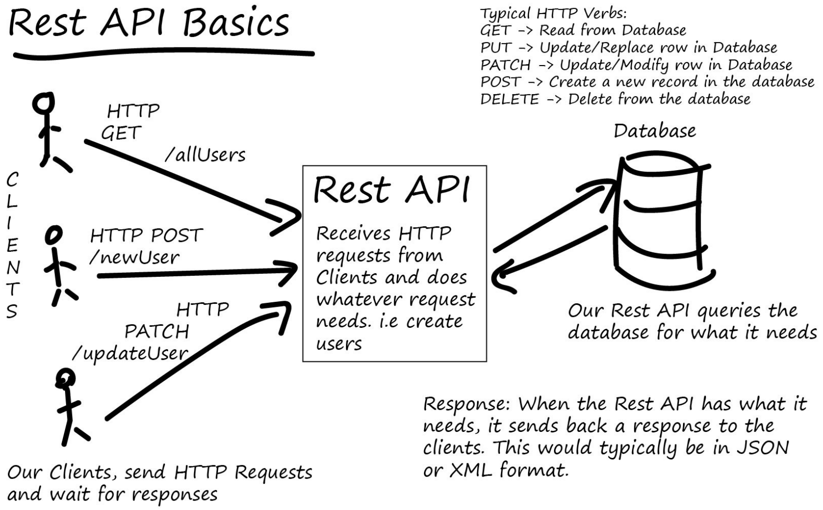
publisher
- idPUBLISHER INT(10)
- name VARCHAR(45)
- location VARCHAR(45)
- Indexes

author
- idAUTHOR INT(10)
- firstName VARCHAR(45)
- lastName VARCHAR(45)
- biography VARCHAR(511)
- Indexes

E-R Diagram

About (JAVA) Skills





Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database

Receives HTTP requests from Clients and does whatever request needs. i.e create users

Our Rest API queries the database for what it needs

Our Clients, send HTTP Requests and wait for responses

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

Teaching Method

The course will be held in ***face to face!***

The teacher will provide in advance (hopefully) the slides used during the lessons (with suggestions to the book chapter to be read).

Video recordings of the last-year classes may be provided.

The course will be held entirely in ***English***.

Tutoring hours (two hours per week) will be provided each ***Monday (REMOTE MODE)*** (17:00-19:00) ***by request***.

Book your meeting (MANDATORY) with teacher here:
https://calendar.app.google/6whAiM6wNSZuxqW58

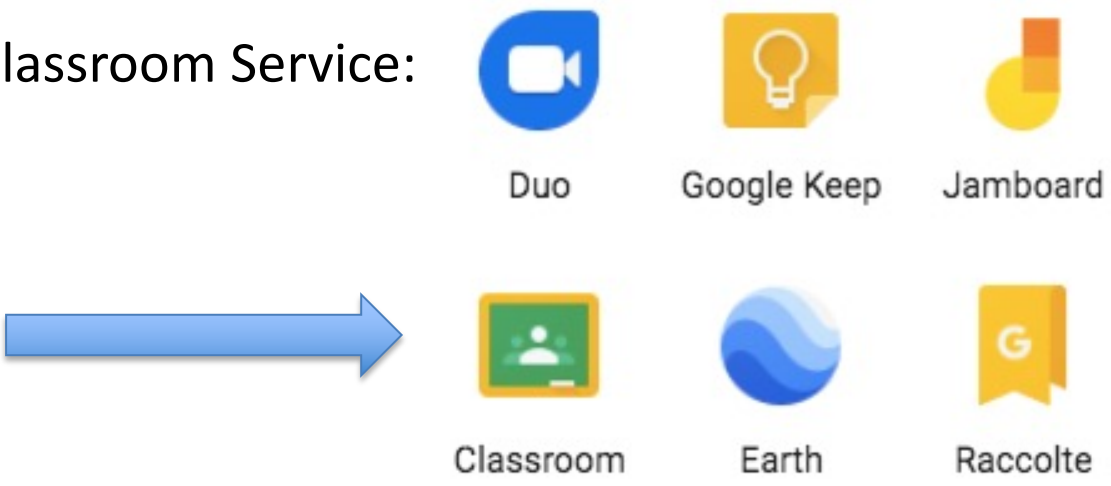
The teacher will be available after the class for a ***Q&A session.***

The E-learning Platform

We will exploit the Google GSuite service for the activities related to the course (materials, tests, projects).

Each student can login to the service with his/her own UNIPI credentials (check details here https://start.unipi.it/en_GB/gsuite/).

Once logged in, select the Classroom Service:



From the + button Join a class (specify Class Code ***zfymq3t***).

Studying Materials

- Slides provided by the Teacher (mostly self-contained)

- Past Videos of the Classes

- Scientific articles provided by the teacher

- Official Documentation of the NoSQL DBMSs.

- Recommended Books:
    - “Guy Harrison, Next Generation Databases, Apress, 2015”
    - “Dan Sullivan, NoSQL For Mere Mortals, Addison-Wesley, 2015”
    - “Andreas Meier, Michael Kaufmann , SQL & NoSQL databases : models, languages, consistency options and architectures for big data management, 2019”
    - “Felipe Cardeneti Mendes, Piotr Sarna, Pavel Emelyanov, Cynthia Dunlop, Database Performance at Scale, Apress Open, 2023”

Check available books at: https://onesearch.unipi.it

THE HISTORY OF THIS COURSE…

2023-2024 A.Y. Experience: Exam Results

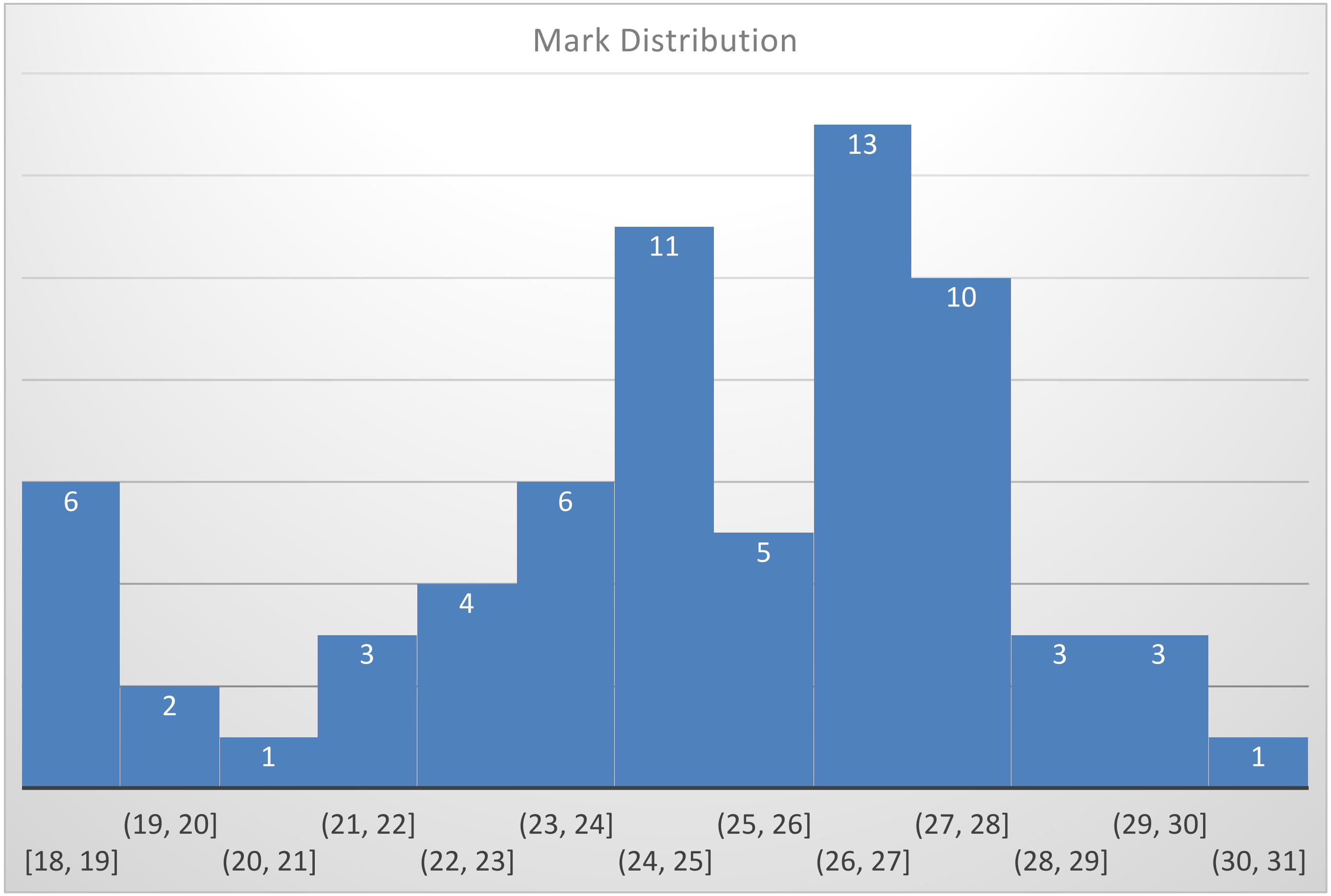
Year	*Code*	*Course*	*Average Mark*	*#Students*
2024	883II	LARGE-SCALE AND MULTI-STRUCTURED DATABASES	25,23	68
2023	883II	LARGE-SCALE AND MULTI-STRUCTURED DATABASES	25,98	63
2022	883II	LARGE-SCALE AND MULTI-STRUCTURED DATABASES	26,99	93
2021	883II	LARGE-SCALE AND MULTI-STRUCTURED DATABASES	27,82	57
2020	883II	LARGE-SCALE AND MULTI-STRUCTURED DATABASES	28,58	83

- The ***percentage*** of project discussed during the first exam session was equal to around 58% (24 groups in total, 14 discussions)

- The ***percentage*** of students attending the course (70) who ***passed the exam*** after the first exam session was equal to ***50% (42.3% in the previous year).***

- The ***percentage*** of students attending the course (70) who ***passed the exam*** was equal to around ***97%***

2023-2024 A.Y. Experience: Exam Results



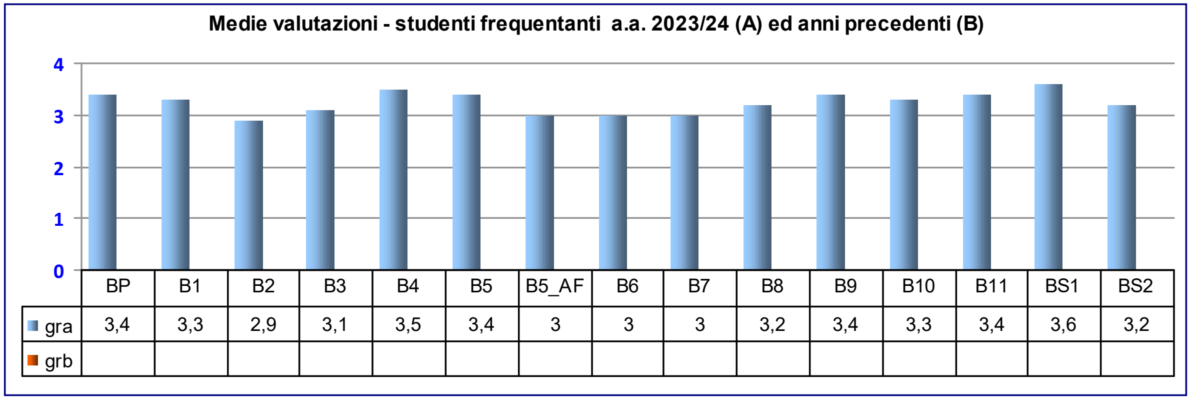
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

2023-2024 A.Y. Experience: Students’ Evaluation (AIDE)

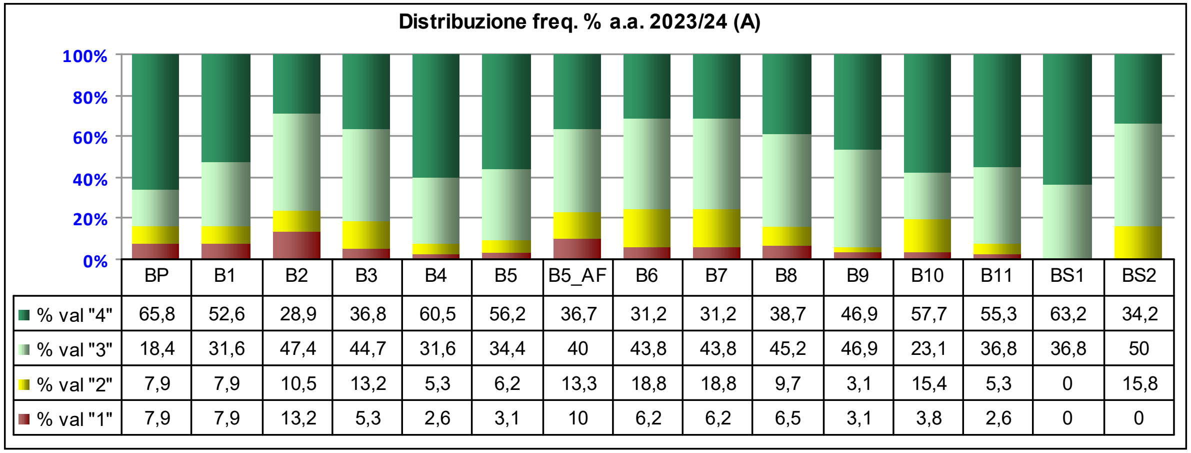
Graf.1



Medie valutazioni - studenti frequentanti a.a. 2023/24 (A) ed anni precedenti (B)

	BP	B1	B2	B3	B4	B5	B5_AF	B6	B7	B8	B9	B10	B11	BS1	BS2
🟦 gra	3,4	3,3	2,9	3,1	3,5	3,4	3	3	3	3,2	3,4	3,3	3,4	3,6	3,2
🟧 grb															

Graf.2



Distribuzione freq. % a.a. 2023/24 (A)

	BP	B1	B2	B3	B4	B5	B5_AF	B6	B7	B8	B9	B10	B11	BS1	BS2
🟩 % val "4"	65,8	52,6	28,9	36,8	60,5	56,2	36,7	31,2	31,2	38,7	46,9	57,7	55,3	63,2	34,2
▨ % val "3"	18,4	31,6	47,4	44,7	31,6	34,4	40	43,8	43,8	45,2	46,9	23,1	36,8	36,8	50
🟨 % val "2"	7,9	7,9	10,5	13,2	5,3	6,2	13,3	18,8	18,8	9,7	3,1	15,4	5,3	0	15,8
🟥 % val "1"	7,9	7,9	13,2	5,3	2,6	3,1	10	6,2	6,2	6,5	3,1	3,8	2,6	0	0

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

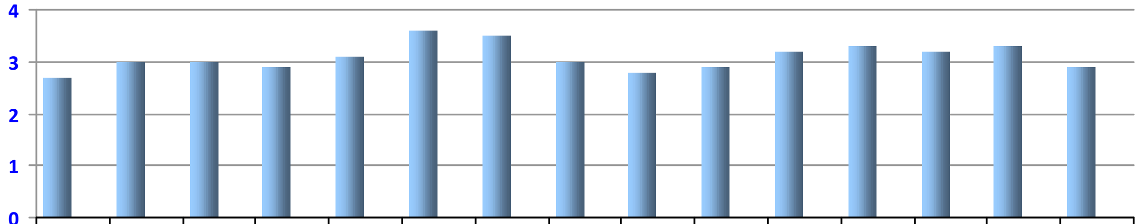
IN SUPREMÆ DIGNITATIS 1343
UNIVERSITÀ DI PISA

CROSSLAB
Innovation for industry 4.0

2023-2024 A.Y. Experience: Students’ Evaluation (CE)

Graf.1

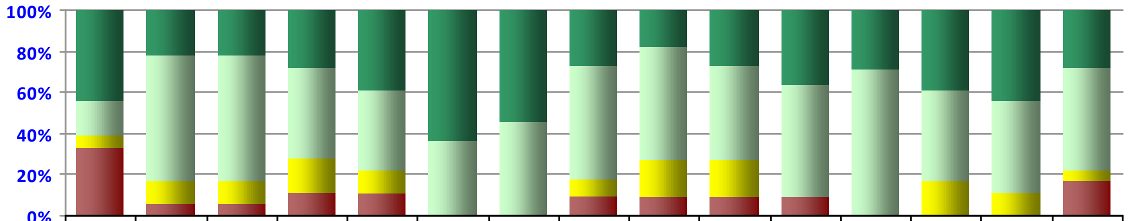
Medie valutazioni - studenti frequentanti a.a. 2023/24 (A) ed anni precedenti (B)



	BP	B1	B2	B3	B4	B5	B5_AF	B6	B7	B8	B9	B10	B11	BS1	BS2
gra	2,7	3	3	2,9	3,1	3,6	3,5	3	2,8	2,9	3,2	3,3	3,2	3,3	2,9
grb															

Graf.2

Distribuzione freq. % a.a. 2023/24 (A)



	BP	B1	B2	B3	B4	B5	B5_AF	B6	B7	B8	B9	B10	B11	BS1	BS2
% val "4"	44,4	22,2	22,2	27,8	38,9	63,6	54,5	27,3	18,2	27,3	36,4	28,6	38,9	44,4	27,8
% val "3"	16,7	61,1	61,1	44,4	38,9	36,4	45,5	54,5	54,5	45,5	54,5	71,4	44,4	44,4	50
% val "2"	5,6	11,1	11,1	16,7	11,1	0	0	9,1	18,2	18,2	0	0	16,7	11,1	5,6
% val "1"	33,3	5,6	5,6	11,1	11,1	0	0	9,1	9,1	9,1	9,1	0	0	0	16,7

DIPARTIMENTO DI INGEGNERIA DELL’INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

2024-2025 Course Implementation

From Now till the mid or end of November 2023

- Classes for introducing the main NoSQL architecture and strategies.
- Recaps of Software Engineering and Java (with exercises)
- Examples of Applications based on NoSQL DB architectures
- Seminars on Best Practices in Developing Java Application (Pending approval)
- Introduction to the main features of NoSQL Infrastructures (Redis, MongoDB, Neo4j)

2023-2024 Course Implementation

Starting from the mid or the end of November 2023

- Introduction to the Project and final Student Group Definition
- Project development

Possible Extra Classes

The week of Dec. 16, 2024 to Dec. 20, 2024 is reserved for any extra class hours as needed (i.e. less than 90 hours delivered).

The Enrolment form

Please fill and submit the following form:

https://forms.gle/2rPaDg5gFGH4VoRR9

It’s mandatory for attending labs and for the cooperative group activities

About the exam

- Discussion of the project (50%)
- Written test on the theoretical parts of the course (50%): three open questions, 30 minutes.

About the Project

Design and develop of an Application interacting with NoSQL Databases

- Start with an idea and a draft of application requirements, use case diagrams and data entities (quick discussion in the classroom during the course or during meeting slots, ***to be approved by the teacher)***

- Refine requirements and use cases

- Define data entities and relationships by means of UML analysis class diagrams

- Provide the design of the Data Base (at least two of the main non relational models must be considered when defining the requirements. ***Document DB model is mandatory***)

- Define the main queries on the Data Base(s)

- Implement and deploy the application, hopefully on the Virtual Lab

- Test the application (by running the implemented RESTful APIs, details will be provided).

- Write a complete documentation resuming the above items and including an APIs documentation.

Some Advices for the Project

- ***Attend carefully*** the lessons during the first 6-8 weeks, especially pay attention to the examples of application discussed by the teacher (often the best projects of past students).

- ***Deepen*** your ***skills*** in Java programming and Software Engineering and make the suggested exercises.

- ***Do not expect*** to receive a ***full coverage*** of all the aspects that may be involved by your projects. ***Spend time*** to check for updates, solutions and news ***by your own***.

- ***Ask support*** to the teacher whenever required.

- Come to the meeting with the teacher for asking for additional clarifications, explanations, advices, and resolving doubts.

Rules for the Project

- The project must be developed ***in groups of 3 students*** (for special cases, talk with the teacher)

- ***No reviews*** are allowed, students will receive a number of examples of past projects.

- The project must be ***discussed before the*** written test (the date will be fixed by the teacher)

- The ***final documentation*** must be submitted to the teacher ***in advance*** (deadline will be fixed some days before the discussion)

- Avoid to ***involve*** the teacher for ***solving problems*** among group members.

Rules for Project Evaluation

Overall Project Evaluation:

- 25 % for the Idea, requirements definitions, the entity-relationship model (and UML diagrams)

- 40% for DB design and query definition

- 25% for the implementation of APIs

- 10% for the clarity of the overall documentation

The ***individual assessment*** will depend on the overall evaluation of the project and the answers given by the specific student during the ***project discussions***.

The Self-Assessment Survey

Each student will be required to fill a survey for ***self-assessing*** their technical and theoretical skills.

https://forms.gle/wC8BnpU2ZFev5QRH8

The results of the survey will be used for ***better focusing the teaching activities***.

The data collected will be used only for ***statistical***, ***teaching*** and ***research*** purposes.

Publications (if any) will only report analysis that will use aggregated and anonymized data .

The ***data will not be transferred*** to third parties or to user profiling companies that may use them for commercial purposes.

If personal data (such as Name, Surname and e-mail address) will be provided, the owner may at ***any time request to delete the data*** and not to use them for further analysis.

Contacts

Prof. Ducange office is located at:

Dipartimento di Ingegneria dell'Informazione, University of Pisa.

Office Address: 1, Largo Lucio Lazzarino, I-56100, Pisa (ITALY)

Room: 4-029

Telephone: +39 050 2217684

EMAIL: pietro.ducange_at_unipi.it (preferred to chatting on MS Teams)

Web: https://sites.google.com/site/ducangepietro

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0