Weekly exercises IN3020&4020 (2025)
Week 12: Security and NoSQL Intro

1 Security

1. Discuss the main issues that we need to deal with in database security as well as the main threats.
2. Introduce discretionary and mandatory security mechanisms for multi-user DBMSs.
3. Discuss main control measures for data security.
4. What is the trade-off between security and precision?
5. What is the difference between security and privacy?
6. Outline the main details of discretionary-based access control (DAC) in SQL.
7. (*) Consider a relation

   $$Employees[Id, Name, Address, Age, Position, Salary].$$

   Write SQL commands with DAC mechanism that give read privileges to *UserA* to the address of the employee with name 'John' as written in the relation (and to nothing else).
8. Outline the main details of mandatory-based access control (MAC) in SQL, including Bell-LaPadula rules.
9. What do we mean by inference control? Give one example where this is important.
10. What is flow control?

2 NoSQL DBMS

1. Name the main motivation points and challenges for NoSQL databases. What do we usually sacrifice in NoSQL to address the challenges in comparison to Relational Databases?

2. What are the V-characteristics of Big Data?
3. What is the CAP theorem, why is this relevant for NoSQL databases?
4. Give the main categories of NoSQL approaches, describe their main ideas and system examples.

3 Try MongoDB (optional)

Setup:

1. To install MongoDB on your computer, follow the instructions for you system here:
   `https://www.mongodb.com/docs/manual/tutorial/`.
2. Copy the file municipalities.json from the web page of the course (same folder where this pdf is)
3. Load data into the database with the command
   `$ mongoload --db counties --collection places --file municipalities.json`
4. After installing and starting the service, you should be able to connect to the database with:
   `$ mongosh`
5. There is a nice overview of operations here: `https://www.mongodb.com/developer/quickstart/cheat-sheet/`.

Try out:

1. Start the shell
2. Show which databases you have by means of operation

   ```
   show dbs
   ```

3. Show which collections you have in the database by means of operation

   ```
   show collections
   ```

4. List all data by means of operation

   ```
   db.places.find()
   ```

5. List data about Akershus by means of operation

   ```
   db.places.find({"county" : "Akershus"})
   ```

6. Remove the entry “Oslo” by means of operation

   ```
   db.places.remove({"county" : "Oslo"})
   ```

7. Insert the entry “Oslo” by means of operation

   ```
   db.places.insert({"county" : "Oslo", "municipalities" : ["Oslo"]})
   ```

8. Find the county with the municipality “Hamar” (do a projection) by means of operation

   ```
   db.places.find({"municipalities": "Hamar"}, {"_id:0", "county":1})
   ```