Weekly exercises IN3020 and IN4020
Week 13

May 5, 2022

1 Security

1. Discuss some of the issues we deal with in database security

   Solution:

   - Legal/ethical
   - Policy
   - System-related Security categorisation/organisation

2. Discuss some threats to databases

   Solution:

   - Loss of integrity
   - Loss of availability
   - Loss of confidentiality

3. What is the trade-off between security and precision?

   Solution: Ideally we would like to have a system with perfect security and maximum precision, this is in contradictory. In practice we need to find the best trad-off between the two. E.g., we want to keep the data sufficiently secure, but we would also like to make it available for use.

4. Discuss *discretionary security* and *mandatory security control (MAC)*. What are the differences, what are the advantages/disadvantages of the two systems.

   Solution:

   - Discretionary security:
     Part of the SQL standard, it has two levels of security. The account level and relation level The account level specifies the access of

the whole database. Data definition/modification/access. While the relational level specifies access to specific tables, with data modification, access and constraints. **Advantages:** It is part of the standard, quite simple **Disadvantages:** Can be challenging to maintain with large systems because it is quite detailed

- Mandatory security control (MAC):
  Used in more "professional" systems. Both relations/tuples and users are classified into security classes. Two rules are enforced (bell-lapadula) simple security and star property. **Advantages:** Easier to control access in large organizations. **Disadvantages:** A bit overkill in smaller systems. Not part of the SQL standard.

5. What do we mean by inference control? Give one example where this is important.

   Solution: Inference control prevents users to infer classified information based on available information. One example is using a statistical database where the user does not have the rights to to access individual data items. They could, however infer such data if they can specify a query that returns the average of one person. Another example is if the user can retrieve some information from a database and then by combining this information with publicly available data recreate sensitive information.

6. What is flow control?

   Solution: Regulation of the flow of information among accessible objects. It ensures that classified information does not flow into less classified objects

2 NoSQL DBMS

1. Name a few reasons why we NoSQL databases (e.g., why can't we just use existing relational databases?)

   Solution:

   - New types of applications with new requirements
   - Distributed systems

2. Name some of the challenges we try to solve using NoSQL databases?

   Solution:

   - Semi structured data
   - High performance

- Data availability
- Scalability
- Sharding and partitioning of data

3. What is the CAP theorem, why is this relevant for NoSQL databases?

   Solution: Relevant because we have a distributed system with replicated data. CAP: If we have a distributed system with replicated data, we cannot at the same time ensure:

   (a) Consistency

   (b) Availability

   (c) Partition tolerance

4. Describe what a document-based database system is. Give one example of such a system

   Solution: It is a collection of similar documents. A document can be complex, self describing and can have different data elements. Typically it is XML or JSON. One example is MongoDB which uses BSON (JSON).

5. Describe what a key-value based database system is and give one example.

   Solution: The database support only key value pairs. This has fast retrieval, high availability, and is scalable. The key is a unique identifier for a data item. The value is the data item itself. E.g., string, array, file. One example is DynamoDB.

6. What is a column-based NoSQL database system? Give an example.

   Solution: A kind of key value based system, but keys are more complex. Apache HBase.

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. Load data into the database with:
   ```
   $ mongoload --db counties --collection places --file municipalities.json
   ```

3. After installing and starting the service, you should be able to connect to the database with:
   `$ mongosh`
4. 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:

   Solution: `show dbs`
3. Show which collections you have in the database:

   Solution: `show collections`
4. List all data

   Solution: `db.places.find()`
5. List data about Akershus

   Solution: `db.places.find({"county" : "Akershus"})`
6. Remove the entry "Oslo"

   Solution: `db.places.remove({"county" : "Oslo"})`
7. Insert the entry "Oslo"

   Solution:
   ```
   db.places.insert({"county" : "Oslo", "municipalities" : ["Oslo"]})
   ```
8. Find the county with the municipality "Hamar" (do a projection)

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

4 Graphs

1. What is a knowledge graph?

   Solution: The definition of a knowledge graph is not universally agreed upon, so you will find some variations. We can say that a knowledge graph is a knowledge base represented using a graphs structure. With a knowledge base one typically means data (e.a., entities or facts) together

with semantic information so that the computer can "interpret" and "understand" the data.

2. Give examples of knowledge graphs technologies?

   Solution:

   - Google knowledge graph
   - Linked data semantic knowledge graphs
   - Neo4j

5 RDF graphs

1. What are some of the W3C certified standards for semantic technologies?

   Solution:

   - data format: RDF
   - ontologies: RDFS/OWL/OWL 2
   - queries: SPARQL
   - (rules: SWRL)
   - (shape constraints: SHACL)
   - ...

2. What is RDF?

   Solution: The Resource Description Framework(RDF) is a framework for data interchange on the Web

3. What is an RDF triple?

   Solution: An atomic data entity in the RDF datamodel consisting of a subject, predicate and object. E.g., <ex:john, ex:father-of, ex:bill>.

4. What is an RDF graph?

   Solution: A set of RDF triples.

5. An RDF graph is often visualized as a directed graph. It is, however, not really a graph. Why?

   Solution: A graph typically consists of two disjoint sets. The edges and the nodes. There is nothing in the definition of an RDF graph that says we cannot express things about a predicate. Consider <rdf:type, rdf:type, rdf:Property> .

6. Why do you think we use URIs to represent concepts in RDF?

   Solution: In relational databases, we need a key to identify a row in a table, RDF need an identifier that is globally unique. URIs are unique across the web. URIs are also addresses so it would be easy to follow them to find data.

7. What is a literal in an RDF graph?

   Solution: Represent unknown information or to represent more complex data.

8. Why do you think we need blank nodes?

9. How would you express the following concepts in RDF?

   (a) IN3020 is a university course

   Solution: There can be many possible solutions here, one example: `<ex:IN3020, rdf:type, ex:UniversityCourse>`

   (b) IN3020 has a lecturer

   Solution: `<ex:IN3020, ex:hasLecturer, _:x>`

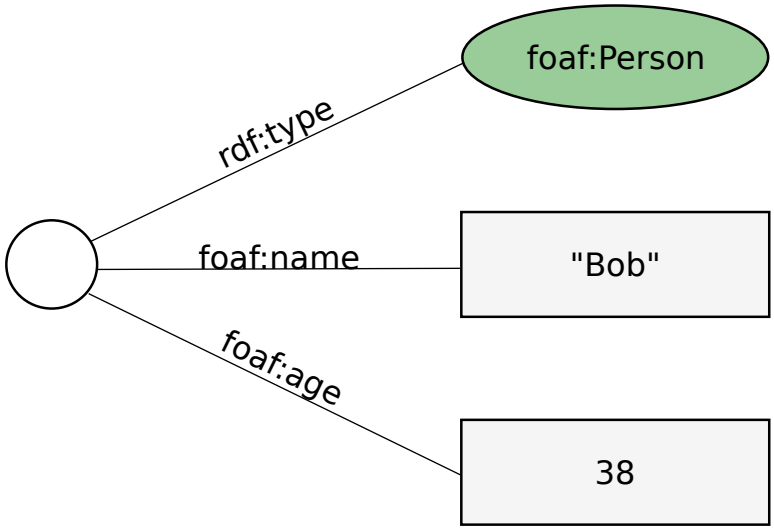
   (c) IN3020 has a lecturer with the name "Egor", he is a person.

   Solution: `<ex:IN3020, ex:hasLecturer, _:x> <_:x foaf:name, "Egor\>`

10. Draw a graph representing the following RDF graph:

```
_:x rdf:type foaf:Person ;
    foaf:name ‘‘Bob’’ ;
    foaf:age   38 .
```

Solution:



11. What is reification?

    Solution: *Dictionary.com:* The act of treating something abstract [...] as if it were a concrete object. In RDF it means to write statements about statements. This way we can overcome the limitation of binary relations and model more complex statements.