KEVIN JOHNSON

MODULE 2 READING COMPREHENSION

1. Define and describe the Relational Model.

The best known data model is SQL, which is based on the relational model. In the relational model, data is organized into relations (tables), where each relation is an unordered collection of tuples (rows). In a relational model, the query optimizer automatically decides which parts of the query to execute in which order, and which indexes to use. The relational model is good for transaction processing and batch processing. It is useful for many-to-many relationships and joins. One criticism of the relational model is if data is stored in relational tables, an awkward translation layer is required between the objects in the application code and the database model of tables, rows, and columns.

2. Define and describe Document Databases.

Document databases have become more common with the adoption of NoSQL. Some driving forces behind adopting NoSQL are a greater need for scalability and specialized query operations. The main arguments in favor of the document data model are schema flexibility, better performance due to locality, and that for some applications it is closer to the data structures used by the application. In the document model, a document is usually stored as a single contiguous string, encoded as JSON, XML, or a binary variant thereof. Document databases take a schema-on-read approach, where the structure of the data is implicit, and only interpreted when the data is read. The schema-on-read approach is advantageous if the items in the collection don’t all have the same structure.

3. Define and describe graph databases.

If many-to-many relationships are very common in your data, it becomes more practical to model your data as a graph. A graph consists on two kinds of objects: vertices and edges.

One example of a graph database is the property graph model. In the property graph model, each vertex consists of a unique identifier, a set of outgoing edges, a set of incoming edges, and a collection of properties (key-value pairs). Each edge consists of a unique identifier, the vertex at which the edge starts (the tail vertex), the vertex at which the edge ends (the head vertex), a label to describe the kind of relationship between the two vertices, and a collection of properties (key-value pairs). Some important aspects of this model are:

1. Any vertex can have an edge connecting it with any other vertex. There is no schema that restricts which kinds of things can or cannot be associated.
2. Given any vertex, you can efficiently find both its incoming and its outgoing edges, and thus traverse the graph both forward and backward.
3. By using different labels for different kinds of relationships, you can store several different kinds of information in a single graph, while still maintaining a clean data model.

The triple-store model is mostly equivalent to the property graph model, using different words to describe the same ideas. In a triple-store, all information is stored in the form of very simple three-part statements: (subject, predicate, object). The subject of a triple is equivalent to a vertex in a graph. The object is one of two things: a value in a primitive datatype such as a string or number, or another vertex in the graph.

4. Discuss the differences between the three different Query Languages covered in this chapter.

MapReduce is a programming model for processing large amount of data in bulk across many machines. MapReduce is neither a declarative query language nor a fully imperative query API, but somewhere in between. The logic of the query is expressed with snippets of code which are called repeatedly by the processing framework. It is based on the map and reduce functions that exist in many functional programming languages.

Cypher is a declarative query language for property graphs. It allows us to state what we want to select, insert, update, or delete from graph data without a description of exactly how to do it. Cypher was based upon the SQL language. However, it is made specifically for graph data and traversals.

SPARQL is a query language for triple-stores using the RDF data model. As a query language, SPARQL is data-oriented in that it only queries the information held in the models. SPARQL does not do anything other than take the description of what the application wants, in the form of a query, and returns that information in the form of a set of bindings or an RDF graph.

5. Discuss the differences between Linked Lists and Arrays from the Algorithms reading.

Using an array means all your tasks are stored contiguously in memory. Arrays are great if you want to read random elements, because you can look up any element in your array instantly. A drawback of arrays if to insert or delete elements in the middle, you have to shift all the other elements in the array. However, arrays still see wide amounts of use became they allow for random access.

With linked lists, your items can be anywhere in memory. Each item stores the address of the next item in the list. Adding an item to a link list is easy, as you stick it anywhere in memory and store the address with the previous item. Inserting or deleting items in the middle of a linked list is easier than arrays because all you have to do is change what the previous element points to. One drawback of linked lists is they only allow for sequential access, meaning reading the elements one by one, starting at the first element.