Algortihms and datastructures

Kristoffer Klokker

2022

Contents

1 Database introduction

Databases are a collection of data stored in a DBMS (database management system) which serves the purpose of:

- Create database and specifying their schemas (logical structure of the data)
- Query the data (questions about data or retrieving the data)
- Store large amount of data in long periods with easy access and modificatio of the data
- Durable and should be able to recover data in case of error or misuse
- Allow multiple user access at once

Today the norm in database systems are relation databasese which present the data as tables, and the underlying datastructure is not needed for use of the system.
In the case of multiple different database and systems which should be syncronised either a data warehouse is used where a periodically copy of the smaller databases is made. Another approach is a middleware which is a translation between two databases schemes.
A database has mainly two users, admin which can modify the schema using DDL command (data-definition language) which modify the schema by altering the metadata.
The other user being a normal user allowed to do DML command (data-manipulation language).
When a DML command is executed two subsystems are handling the command:

1.1 Query compiler

The compiler takes the query and creates a query plan (a sequence of actions) and passes it the the execution engine.
A request of data sends data data in tuples to the buffer manager, which is responsible for all data transaction between disk storage and memory
The compiler consist of

- Query parser - which builds a tree from the textual query
- Query preprocessor - Sematic check of query to ensure a valid query and transforms the query into algebraic operators

- Query optimizer - Transofrm the query to the best avaliable sequence of operation on the actual data based on metadata and schema structure

1.2 Transaction manager

The transaction manager is used to log for possible recovering and ensuring durablity
Also the transaction has a concurrency-controle manager to ensure a bundle of transaction is executed as they were one unit and locking data when used to ensure no data is wrongly overwritten.
The transaction also manages such that every execution is isolated in case of revertion.
The transaction followed the ACID test, where

- A - atomicity which ensures every execution is done at 1 step at the time
- C - consistency in data and data constraints
- I - isolation of every transaction
- D - durability of data such it is never lost after a transaction

2 The relational model of data

A data model is used for describing data and conisist of:

- Structure of data - Referred to as physical data model, but is simply a high level data structure
- Operations on the data - A limited set of operations in DBS at hight level, which makes it more flexible for underlying improvements
- Constraints of data - Constraints on data to ensure data integrity

2.1 The semistructured-data model

The data is setup in a relation more like a tree rather than table.
Here XML is mostly used to represent datam by nested tags.

```
<Movies>
    <Movie title="Gone with the wind">
        <Year>1939</Year>
```

```
        <Length>231</Length>
        <Genre>drama</Genre>
    </Movie>
    <Movie title="Star Wars">
        <Year>1977</Year>
        <Length>124</Length>
        <Genre>sciFi</Genre>
    </Movie>
</Movies>
```

2.2 The basics of relational database

Relation refers to the two dimensional table of data.With attributes being the coloumns and rows being a tuple. The tuple is then made of an relations where a relation with attributes are a schema.
A relation is defined by $Name(attribute : type, attribute2 : type)$ and a tuple is in the same order and valeis for the given attributes.
Relations comes in sets and not lists and therefore order is not important
A database may contain a key which is attribute(s) which define a unique relation, if no combination of attributes are unique a ID for the relation can be created.

2.3 SQL language

SQL is the language used to create queries. SQL has tree kinds of relations, stored called tables (relations), views (relation which are not stored but used for computation), temporary tables (tables constructed by SQL temporary)
The data types avaliable by SQL are:

- $CHAR(n)$ - Character string of fixed length n
- BIT - Logical value with possible values being TRUE, FALSE, UNKNOWN
- INT - Number can also be $SHORTINT$ for small number
- $FLOAT$ - Higher precision numbers here $DOUBLE$ can also be used for more precision
- $DECIMAL(n, d)$ - Numbers of length n and the decinam placed at d
- $DATE$ and $TIME$ - both essentially being strings with a strict format

The basic commands for modifying tables are:

- $DROPTABLER$; which removes the table R with all its entries
- $ALTERTABLERADDdatype$ Adds attribute a as a *type* to table R
- $ALTERTALBERDROPa$ Removes the attribute a form table R

SQL also has $DEFAULT$ which can be added after any attribute after type and describes the default value if non is given.

2.3.1 Keys

A PRIMARY KEY is used for securing no dublicates and only allows non null values in the key attribute.
UNIQUE allows null as a value in its attribute, but dublicates is still nto allowed.
When creating a table the key can be choosen by after an attribute after its type $PRIMARYKEY$ or $UNIQUE$ is inserted or at the end of the table definition $PRIMARYKEY(a)$ can be inserted where a are the attributes. Again Unique can also be used like this.