When considering the correct database for our project, we had the option of choosing between three options: the standard relational databases, non-relational databases or go with a left field choice but which is radically gaining popularity, blockchain. To come to a final decision, we have decided to list the pros and cons of each solution to make an informed final decision to see which approach fits our needs the most.

There are two main types of database technologies, the first being Relational Databases.

Relational databases are great at organizing and retrieving structured data

**Pros**

* Relational databases work with structured data.
* They support ACID transactional consistency and support “joins.”
* They come with built-in data integrity and a large eco-system.
* Relationships in this system have constraints.
* There is limitless indexing. Strong SQL.

**Cons**

* Relational Databases do not scale out horizontally very well, only vertically.
* Data is normalized, meaning lots of joins, which affects speed.
* They have problems working with semi-structured data.

Non-relational databases or NoSQL are best applicable when data is inconsistent, incomplete or its amount is massive, this approach is very popular in the bigdata field.

**Pros**

* They scale out horizontally and work with unstructured and semi-structured data.
* Schema-free or Schema-on-read options.
* High availability.
* Many NoSQL databases are open source and so “free”.

**Cons**

* Weaker or eventual consistency (BASE) instead of ACID.
* Limited support for joins.
* Data is denormalized, requiring mass updates.
* Does not have built-in data integrity.
* Limited indexing.
* Requires considerable training, to be used properly.

Blockchain, while a radically different approach to the traditional SQL/NoSQL solution it would be interesting to see how this innovative way of storing data could bypass the need for a future system administrator and open up possibilities to create decentralized applications or Dapps for short, which members of GTL could use to track the status of books or even be incentivized to borrow books in the form of rewards.

**Pros**

* Transactions are immutable
* All transactions are linked to one and other and are stored forever, great for the statistics/analysis requirement given by GTL
* High availability
* Increased transparency over traditional databases

**Cons**

* Can be costly depending on the type of implementation
* Low interoperability with traditional databases
* No experience with the technology