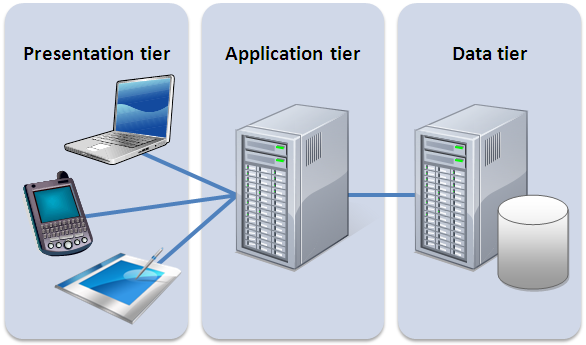
**Software architecture (high-level/architectural design + main design decisions)**

Architecture style = use the three-tier architecture (combination of the layered and client-server architecture)



Source (picture): <https://managementmania.com/en/three-tier-architecture>

Presentation tier: HTML, CSS, Bootstrap

Application tier: Javascript, AngularJS, PHP, Ajax

Data tier: MySQL ([www.phpmyadmin.net](http://www.phpmyadmin.net))

Pros of using the three-tier architecture:

* **More secure** – full control over each particular tier’s security (some might require more security than others like the server and database)
* **Easily scalable** – upgrade individual tiers without affecting the other tiers
* **Easily maintained** – you can make changes to a particular tier without affecting the other theirs

Source (video about pros): <https://www.youtube.com/watch?v=KlHvRKSH4pk>

**Motivation of the design choices and how the client’s requirements have affected the design choices**

* The three-tier architecture is commonly used for web-applications. So it is a well-tested architecture in real life scenarios.
* We needed a database because the client wanted the following requirements:

1. **Update the default input vales, guiding texts etc. by uploading an Excel file** (needed to store all this data somewhere for the users to be able to retrieve it when using the website)
2. **Switch language** (one column for Swedish, and one column for the English – easy way)
3. **Save calculated results** (need to be able to save the previous results efficiently somewhere)
4. **Compare two results** (need to retrieve the previously stored results for comparison)

**Alternative for database = cookies:**

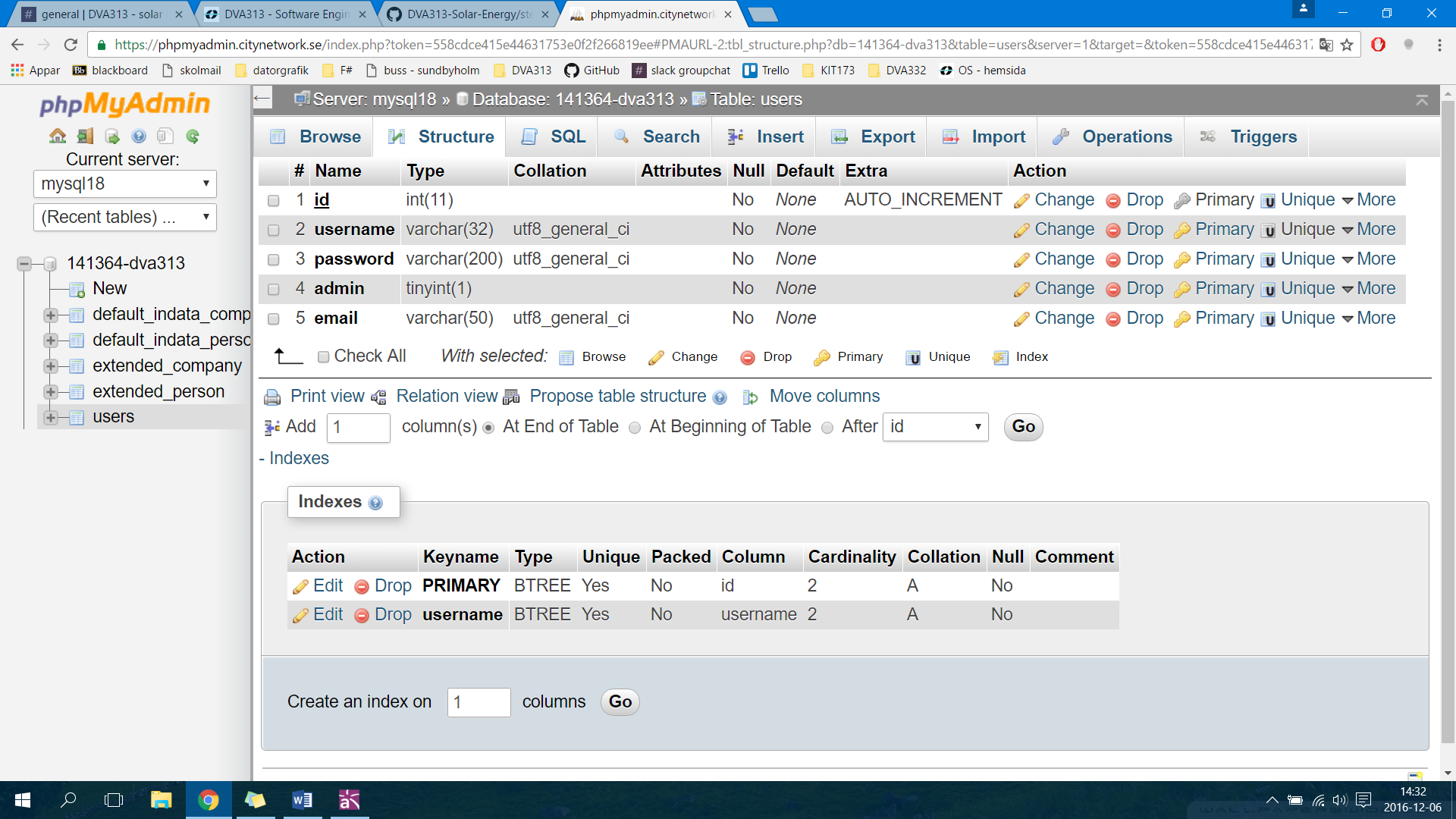
The reason why we chose to use a database instead of cookies when saving inputted data, calculated results etc.

=

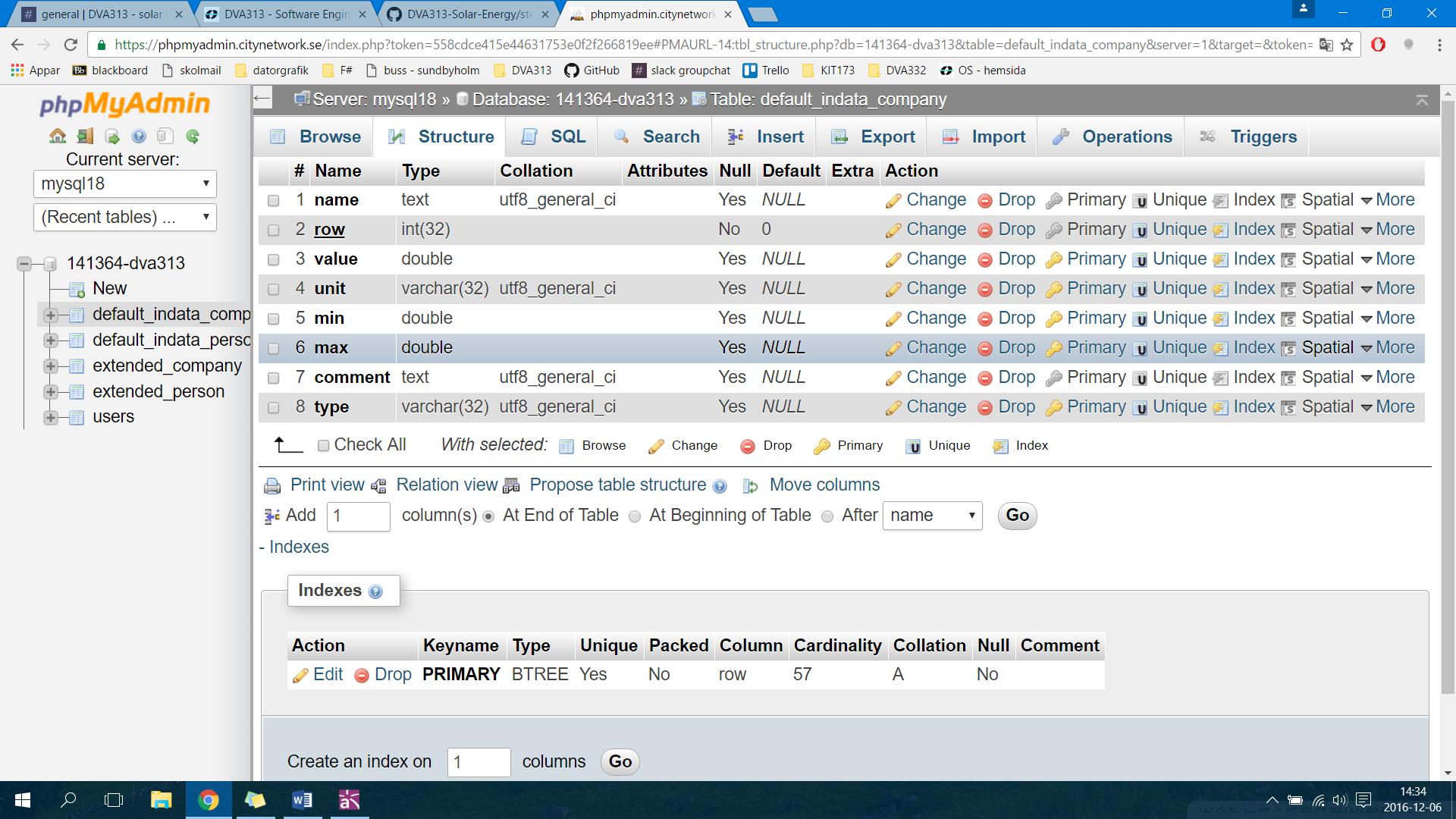
if you use a different computer or if you deleted your cookies, then all of your previously inputted/saved data will not be available. It is also extremely annoying to re-enter all of the several input values.

**Detailed design (what the available database tables look like right now)**

We have one database table for all of the users’ login data:



We have one database table each for the **default input data** of both private persons and others (e.g. companies) – which have the same field structure:



**OBS.** the database tables for the **extended input data** looks the same as the one above. (one each for private persons and others (e.g. companies).